RL-TR-91-407, Vol II (of two)
Final Technical Report
December 1991

QUALITY EVALUATION SYSTEM (QUES)
Software Quality Framework as Implemented in QUES

Software Productivity Solutions, Inc.

Karen A. Dyson

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-407, Volume II (of two) has been reviewed and is approved for publication.

APPROVED: Roger J. Dziegiel, Jr

ROGER J. DZIEGIEL, JR.
Project Engineer

FOR THE COMMANDER: John A Graniero

JOHN A. GRANIERO
Chief Scientist
Command, Control, and Communications

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RLC3CB) Griffiss AFB, NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
December 1991

3. REPORT TYPE AND DATES COVERED
Final Dec 87 - Sep 91

4. TITLE AND SUBTITLE
QUALITY EVALUATION SYSTEM (QUES),
Software Quality Framework as Implemented in QUES

5. FUNDING NUMBERS
C - F30602-88-C-0019
PE - 63728F
PR - 2527
TA - 03
WU - 15

6. AUTHOR(S)
Karen A. Dyson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Software Productivity Solutions, Inc.
122 N. 4th Avenue
Indialantic FL 32903

8. PERFORMING ORGANIZATION REPORT NUMBER
A015

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3CB)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-407, Vol II
(of two)

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Roger J. Dziegiel, Jr/C3CB/(315) 330-4476

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Volume I presents the features of the Quality Evaluation System (QUES). QUES is a tool which automates the process of establishing, maintaining, and applying a software quality evaluation framework. Features are illustrated with examples and reports generated by the tool. Main features include framework creation, framework tailoring, project creation and data collection. QUES will support analysis of Fortran and Ada code on a DEC VAX and analysis of Ada code on a SUN SparcStation.

Volume II summarizes the Rome Laboratory Software Quality Framework(RLSQF) as automated with the QUES tool. This volume upgrades the original RADC Software Quality Framework documented in 1985. The primary changes were improved objectivity of questions, improved automatability of questions, improved applicability to Ada, and consistency with DOD-STD-2167A lifecycle phases and terminology.

14. SUBJECT TERMS
Software Quality, Software Evaluation, Metrics.
Software Quality Indicators

15. NUMBER OF PAGES
280

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

1.0 INTRODUCTION

1.1 Software Quality Framework

Since 1976, the Rome Laboratory (RL), formerly known as the Rome Air Development Center (RADC) has pursued a program intended to achieve better control of software quality. Through a series of contracts, this program has sought to identify key software quality issues and to provide a valid methodology for specifying and measuring software quality levels.

A quality model was established in which a hierarchical relationship exists between a user-oriented quality *factor* at the top level, and software-oriented attributes at the second and third levels (*criteria* and *metrics*, respectively). This model is shown in Figure 1.1-1. Software quality is predicted and measured by the presence, absence, or degree of identifiable software attributes.

The model is flexible. It shows the general relationship between each factor and its attributes, and can be updated without effecting the structure of the model itself. This version of the Software Quality Framework, representing this model, is the most recent updating of contents and relationships. It is intended to serve as a baseline for further development and research, and is to be used by tools that support the framework

The metric elements (the specific questions applied to a project under development to assess and predict quality) are the key element of this report. These metric elements are listed in section 2. Each Data Collection Form (DCF) in that section is organized to correspond to a phase of the development process. These phases and the associated Data Collection Form are:

- System Requirements Analysis/Design -- DCF A
- Software Requirements Analysis -- DCF B
- Preliminary Design -- DCF C
- Detailed Design -- DCF D
- Coding and CSU Testing -- DCF E
- CSC Integration and Test -- DCF F
- CSCI Testing -- DCF G
- System Testing -- DCF H
- Operational Test and Evaluation -- DCF I

In addition, information is gathered at various levels of the software implementation architecture. Information can be collected based on system-wide questions, on questions applicable to each Computer Software Configuration Item (CSCI), questions applicable to each Computer Software Component (CSC), and questions applicable to each Computer Software Unit (CSU). These terms are used to represent the various generic software architecture levels, and may be defined by the user to apply to various components of his particular system under development.

The methodology used to apply these metrics is currently defined in the following guidebook:

- *Specification of Software Quality Attributes -- Software Quality Specification Guidebook.* Volume II. Rome Air Development Center. October, 1984.
- *Specification of Software Quality Attributes -- Software Quality Evaluation Guidebook.* Volume III. Rome Air Development Center. October, 1984.

The Software Quality Framework currently consists of 13 factors, shown in Table 1.1-1. Each of these factors is associated with criteria as defined in Table 1.1-2. The relationship between the criteria and factors is shown in Table 1.1-3. Specific software attributes called metrics are associated with the criteria. Table 1.1-4 lists each metric.

1.2 Report Organization

Section 2 of this report lists the Data Collection forms that contain the individual metric elements for each phase of the system life cycle.

Section 3 of the report describes the process of scoring the metric elements, once the data has been collected.

Table 1.1-1. SOFTWARE QUALITY FRAMEWORK FACTORS

Efficiency	Deals with the utilization of a resource
Integrity	Deals with software security failures due to unauthorized access
Reliability	Deals with software failures
Survivability	Deals with software continuing to perform when a portion of the system has failed
Usability	Deals with the relative effort involved in learning about and using software
Correctness	Deals with the extent to which software design and implementation conforms to specification and standards
Maintainability	Deals with the ease of effort in locating and fixing software failures
Verifiability	Deals with software design characteristics affecting the effort to verify software operation and performance
Expandability	Deals with the relative effort in increasing software capabilities or performance
Flexibility	Deals with the ease of effort in changing software to accommodate changes in requirements
Interoperability	Deals with the relative effort in coupling software of one system to software of one or more other systems
Portability	Deals with the relative effort involved in transporting software to another environment
Reusability	Deals with the relative effort for converting a portion of software for use in another application

Table 1.1-2. SOFTWARE QUALITY FRAMEWORK CRITERIA

Accuracy	Those characteristics of software which provide the required precision in calculations and output.
Anomaly Management	Those characteristics of software which provide for continuity of operations under, and recovery from, non-nominal conditions
Autonomy	Those characteristics of software which determine its non-dependency on interfaces and functions
Distributedness	Those characteristics of software which determine the degree to which software functions are geographically or logically separated within the system
Effectiveness-Communication	Those characteristics of the software which provide for minimum utilization of communication resources in performing functions
Effectiveness-Processing	Those characteristics of software which provide for minimum utilization of processing resources in performing functions
Effectiveness-Storage	Those characteristics of the software which provide for minimum utilization of storage resources
Operability	Those characteristics of software which determine operations and procedures concerned with operation of software and which provide useful inputs and outputs which can be assimilated
Reconfigurability	Those characteristics of software which provide for continuity of system operation when one or more processors, storage units, or communication links fails

Table 1.1-2. SOFTWARE QUALITY FRAMEWORK CRITERIA (Continued)

System Accessibility	Those characteristics of software which provide for control and audit of access to the software and data
Training	Those characteristics of software which provide transition from current operation and provide initial familiarization
Completeness	Those characteristics of software which provide full implementation of the functions required
Consistency	Those characteristics of software which provide for uniform design and implementation techniques and notation
Traceability	Those characteristics of software which provide a thread of origin from the implementation to the requirements with respect to the specified development envelope and operational environment
Visibility	Those characteristics of software which provide status monitoring of the development and operation
Application Independence	Those characteristics of software which determine its nondependency on database system, microcode, computer architecture, and algorithms
Augmentability	Those characteristics of software which provide for expansion of capability for functions and data
Commonality	Those characteristics of software which provide for the use of interface standards for protocols, routines, and data representations
Document Accessibility	Those characteristics of software which provide for easy access to software and selective use of its components

Table 1.1-2. SOFTWARE QUALITY FRAMEWORK CRITERIA
(Continued)

Functional Overlap	Those characteristics of software which provide common functions to multiple systems
Functional Scope	Those characteristics of software which provide commonality of functions among applications
Generality	Those characteristics of software which provide breadth to the functions performed with respect to the application
Independence	Those characteristics of software which determine its non-dependency on software environment (computing system, operating system , utilities, input/output routines, libraries)
System Clarity	Those characteristics of software which provide for clear description of program structure in a non-complex and understandable manner
System Compatibility	Those characteristics of software which provide the hardware, software, and communication compatibility of two systems
Virtuality	Those characteristics of software which present a system that does not require user knowledge of the physical, logical, or topological characteristics
Modularity	Those characteristics of software which provide a structure of highly cohesive components with optimum coupling
Self-Descriptiveness	Those characteristics of the software which provide explanation of the implementation of functions
Simplicity	Those characteristics of software which provide for definition and implementation of functions in the most noncomplex and understandable manner

Table 1.1-3. SOFTWARE QUALITY FRAMEWORK FACTORS AND ASSOCIATED CRITERIA

FACTOR Criterion	Acronym	EFFICIENCY	INTEGRITY	RELIABILITY	SURVIVABILITY	USABILITY	CORRECTNESS	MAINTAINABILITY	VERIFIABILITY	EXPANDABILITY	FLEXIBILITY	INTEROPERABILITY	PORTABILITY	REUSABILITY
Accuracy	AC			x										
Anomaly Management	AM			x	x									
Autonomy	AU				x									
Distributedness	DI				x									
Effectiveness - Communication	EC	x												
Effectiveness - Processing	EP	x												
Effectiveness - Storage	ES	x												
Operability	OP					x								
Reconfigurability	RE				x									
System Accessibility	SS		x											
Training	TN					x								
Completeness	CP						x							
Consistency	CS						x	x						
Traceability	TC						x							
Visibility	VS							x	x					
Application Independence	AP													x
Augmentability	AT									x				
Commonality	CL											x		
Document Accessibility	DO													x
Functional Overlap	FO											x		
Functional Scope	FS													x
Generality	GE									x	x			x
Independence	ID											x	x	x
System Clarity	ST													x
System Compatibility	SY											x		
Virtuality	VR									x				
Modularity	MO				x			x	x	x	x	x	x	x
Self-Descriptiveness	SD							x	x	x	x		x	x
Simplicity	SI			x				x	x	x	x			x

Table 1.1-4. SOFTWARE QUALITY FRAMEWORK METRICS

Accuracy	AC.1	Accuracy
Anomaly Management	AM.1	Error Tolerance/Control
	AM.2	Improper Input Data
	AM.3	Computational Failures
	AM.4	Hardware Faults
	AM.5	Device Errors
	AM.6	Communications Errors
	AM.7	Node/Communications Errors
Application Independence	AP.1	Database Management Implementation Independence
	AP.2	Data Structure
	AP.3	Architecture Standardization
	AP.4	Microcode Independence
	AP.5	Functional Independence
Augmentability	AT.1	Data Storage Expansion
	AT.2	Computation Extensibility
	AT.3	Channel Extensibility
	AT.4	Design Extensibility
Autonomy	AU.1	Interface Complexity
	AU.2	Self-Sufficiency
Commonality	CL.1	Communications Commonality
	CL.2	Data Commonality
	CL.3	Common Vocabulary
Completeness	CP.1	Completeness
Consistency	CS.1	Procedure Consistency
	CS.2	Data Consistency
Distributedness	DI.1	Design Structure

Table 1.1-4. SOFTWARE QUALITY FRAMEWORK METRICS
(Continued)

Document Accessibility	DO.1	Access to Documentation
	DO.2	Well-Structured Documentation
Effectiveness--Communication	EC.1	Communication Effectiveness
Effectiveness--Processing	EP.1	Processing Effectiveness
	EP.2	Data Usage Effectiveness
Effectiveness--Storage	ES.1	Storage Effectiveness
Functional Overlap	FO.1	Functional Overlap
Functional Scope	FS.1	Function Specificity
	FS.2	Function Commonality
	FS.3	Function Selective Usability
Generality	GE.1	Unit Referencing
	GE.2	Unit Implementation
Independence	ID.1	Software Independence from System
	ID.2	Machine Independence
Modularity	MO.1	Modular Implementation
	MO.2	Modular Design
Operability	OP.1	Operability
	OP.2	User Input Communicativeness
	OP.3	User Output Communicativeness
Reconfigurability	RE.1	Reconfigurability
Self-Descriptiveness	SD.1	Quantity of Comments
	SD.2	Effectiveness of Comments
	SD.3	Descriptiveness of Language

Table 1.1-4. SOFTWARE QUALITY FRAMEWORK METRICS (Continued)

Simplicity	SI.1	Design Structure
	SI.2	Structured Language or Preprocessor
	SI.3	Data and Control Flow Complexity
	SI.4	Coding Simplicity
	SI.5	Specificity
	SI.6	Halstead's Level of Difficulty Measure
System Accessibility	SS.1	Access Control
	SS.2	Access Audit
System Clarity	ST.1	Interface Complexity
	ST.2	Program Flow Complexity
	ST.3	Application Functional Complexity
	ST.4	Communication Complexity
	ST.5	Structure Clarity
System Compatibility	SY.1	Communication Compatibility
	SY.2	Data Compatibility
	SY.3	Hardware Compatibility
	SY.4	Software Compatibility
	SY.5	Documentation for Other System
Traceability	TC.1	Cross Reference
Training	TN.1	Training
Virtuality	VR.1	System/Data Independence
Visibility	VS.1	Unit Testing
	VS.2	Integration Testing
	VS.3	CSCI Testing
	VS.4	Functional Testing

2.0 SOFTWARE QUALITY DATA COLLECTION FORMS

This section of the report contains the Software Quality Framework Data Collection Forms.

DATA COLLECTION FORM -- A-LEVEL

DCFA-1/DCFA-2

AC.1.1.a (SYSTEM)

Is there a reference to available documentation which describes the results of an error analysis?

Y/N/NA

AC.1.2.a (SYSTEM)

Have accuracy requirements been budgeted to the individual capabilities?

Y/N/NA

AC.1.4.a (SYSTEM)

Are there quantitative accuracy requirements for all applicable inputs associated with each mission critical capability?

Y/N/NA

AC.1.5.a (SYSTEM)

Are there quantitative accuracy requirements for all applicable outputs associated with each mission critical capability?

Y/N/NA

AC.1.6.a (SYSTEM)

Are there quantitative accuracy requirements for all applicable constants associated with each mission critical capability?

Y/N/NA

AC.1.7.a (SYSTEM)

Do the existing math library subprograms which are planned for use provide enough precision to support accuracy objectives?

Y/N/NA

AM.1.1.a (SYSTEM)

How many instances are there of different processes (capabilities, constituent capabilities) allowed to execute at the same time (e.g., concurrent processing, tasking)?

__/NA

AM.1.2.a (SYSTEM)

In how many instances is concurrent processing or tasking centrally controlled?

__/NA

AM.1.3.a (SYSTEM)

How many error conditions are identified?

__/NA

AM.1.4.a (SYSTEM)

How many recognized error conditions require recovery or repair of the error?

__/NA

AM.1.5.a (SYSTEM)

Is there a standard for handling recognized errors such that all error conditions are reported (via raising, propagating exceptions, or passing a value) to the calling body (e.g., subprogram, task, or package)?

Y/N/NA

AM.1.6.a (SYSTEM)

How many instances are there of the same process (capability, constituent capability) being required to execute more than once for comparison purposes? (For example, polling of parallel processing, redundant processing or tasking results.)

__/NA

AM.1.7.a (SYSTEM)

How many instances of redundant processing are required to be centrally controlled?

__/NA

AM.2.1.a (SYSTEM)

Are error tolerances specified for particular external input data? (For example, range of numerical values, legal combinations of alphanumeric values.)

Y/N/NA

AM.3.1.a (SYSTEM)

Are there requirements for detection of and recovery from all computational failures (i.e., using exception handlers or other means)?

Y/N/NA

AM.4.1.a (SYSTEM)

Are there requirements to recover (i.e., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, power failure, clock interrupt)?

Y/N/NA

AM.5.1.a (SYSTEM)

Are there requirements to recover from all I/O device errors (i.e., I/O Exception - Device Error)?

Y/N/NA

AM.6.1.a (SYSTEM)

Are there requirements to recover (i.e., through exception handlers or other means) from all communication transmission errors?

Y/N/NA

AM.7.1.a (SYSTEM)

Are there requirements to recover from all failures to communicate with other nodes/systems?

Y/N/NA

AM.7.2.a (SYSTEM)

Is there a requirement to periodically check adjacent nodes or interoperating systems for operational status?

Y/N/NA

AM.7.3.a (SYSTEM)

Is there a requirement to provide a strategy for alternate routing of messages?

Y/N/NA

AP.1.1.a (SYSTEM)

Is there a requirement to limit specific references to the database management scheme? (For example, all data calls to database information are processed through an executive CSCI.)

Y/N/NA

AP.2.3.a (SYSTEM)

Is there a standard for commenting all global data within a CSU to show where the data is derived, the data's composition, and how the data is used?

Y/N/NA

AP.2.4.a (SYSTEM)

Is there a standard for commenting all parameter input/output and local variables in CSUs which includes the data's composition and use?

Y/N/NA

AP.3.1.a (SYSTEM)

Is there a requirement to localize specific references to computer architecture, (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.a (SYSTEM)

Is there a requirement to avoid or limit the use of microcode instruction statements?

Y/N/NA

AP.5.1.a (SYSTEM)

Is there a requirement to develop functional processing algorithms which are not unique to this system's application (i.e., can be used in similar applications with minimal tailoring)?

Y/N/NA

AP.5.2.a (SYSTEM)

How many system capabilities are there?

__/NA

AP.5.3.a (SYSTEM)

How many capabilities are there with algorithms which were reused from other system applications?

__/NA

AT.1.2.a (SYSTEM)

Are there requirements for spare memory storage capacity?

Y/N/NA

AT.1.5.a (SYSTEM)

Are there requirements for spare auxiliary storage capacity?

Y/N/NA

AT.2.3.a (SYSTEM)

Are there requirements for spare processing capacity (time)?

Y/N/NA

AT.3.1.a (SYSTEM)

Are there requirements for spare I/O channel capacity (time)?

Y/N/NA

AT.3.4.a (SYSTEM)

Are there requirements for spare communication channel capacity (time)?

Y/N/NA

AT.4.1.a (SYSTEM)

Are there requirements for interface compatibility among all processors, communication links, memory devices, and peripherals within the system?

Y/N/NA

AT.4.2.a (SYSTEM)

Is documentation available which describes the results of previous engineering studies (such as trade-off studies, feasibility studies, risk analysis, requirement definitions)?

Y/N/NA

AT.4.3.a (SYSTEM)

Is documentation available which describes new or emerging software areas which may affect the scope of the software requirements or the software implementation techniques? (For example, voice recognition using AI techniques.)

Y/N/NA

AU.2.1.a (SYSTEM)

Are there requirements for each CPU/SYSTEM to have a separate power source?

Y/N/NA

AU.2.2.a (SYSTEM)

Are there requirements for executive software to test its own operation, communication links, memory devices, and peripheral devices?

Y/N/NA

CL.1.1.a (SYSTEM)

Are there requirements for communication with other systems?

Y/N/NA

CL.1.2.a (SYSTEM)

Is there requirement for protocol standards to control all network communications?

Y/N/NA

CL.1.3.a (SYSTEM)

How many system capabilities receive input from other systems?

__/NA

CL.1.4.a (SYSTEM)

How many system capabilities transmit output to other systems?

__/NA

CL.1.5.a (SYSTEM)

Are there requirements to control all network processing?

Y/N/NA

CL.1.6.a (SYSTEM)

Is network processing control part of the protocol standards?

Y/N/NA

CL.1.7.a (SYSTEM)

Are there requirements to control all user sessions?

Y/N/NA

CL.1.8,a (SYSTEM)

Is user session control part of the network protocol standards?

Y/N/NA

CL.1.9.a (SYSTEM)

Are there requirements for a communication routing strategy?

Y/N/NA

CL.1.10.a (SYSTEM)

Is a communication routing strategy part of the network protocol standards?

Y/N/NA

CL.1.11.a (SYSTEM)

Are there requirements to handle messages (e.g., synchronization, message decoding) in an uniform manner?

Y/N/NA

CL.1.12.a (SYSTEM)

Is uniform message handling (e.g. synchronization, message decoding) part of the network protocol standards?

Y/N/NA

CL.1.13.a (SYSTEM)

How many other systems must respond correctly to successfully complete synchronization?

__/NA

CL.1.14.a (SYSTEM)

Is the system free from synchronization imposing constraints upon system computation or response time (e.g., resulting in a user wait time of more than several seconds)?

Y/N/NA

CL.1.15.a (SYSTEM)
Is the system free from time-critical constraints with respect to external communication? (For example, data freshness.)

Y/N/NA

CL.1.16.a (SYSTEM)
How many other systems is this system required to interface with?

__/NA

CL.1.17.a (SYSTEM)
Is there a description of how the computer system will appear to its users and how the users will interact with the system?

Y/N/NA

CL.1.18.a (SYSTEM)
Is there a complete and definitive set of operating procedures for using this system?

Y/N/NA

CL.2.1.a (SYSTEM)
Are there requirements for a standard to establish common representations of data for uniform communications with other systems?

Y/N/NA

CL.2.2.a (SYSTEM)
Has a standard been established for common representations of data and/or for translations between representations of data for uniform communications with other systems?

Y/N/NA

CL.2.3.a (SYSTEM)
How many system capabilities perform data translations?

__/NA

CL.2.4.a (SYSTEM)
Is there a requirement to receive all input data from other systems in common formats? (For example, common format for data positioning, data packing, block transmission.)

Y/N/NA

CL.2.5.a (SYSTEM)
How many different formats are used for input data from other systems (e.g., formats for data positioning, data packing, block transmission)?

__/NA

CL.2.6.a (SYSTEM)

Is there a requirement to output all data to other systems in common formats (e.g., formats for data positioning, data packing, block transmission)?

Y/N/NA

CL.2.7.a (SYSTEM)

How many different formats are used to output data to other systems (e.g., formats for data positioning, data packing, block transmission)?

__/NA

CL.2.8.a (SYSTEM)

How many different types of input records are received from other systems?

__/NA

CL.2.9.a (SYSTEM)

How many types of input records transmitted from other systems contain tags identifying the type of data they contain?

__/NA

CL.2.10.a (SYSTEM)

How many different types of output records are transmitted to other systems?

__/NA

CL.2.11.a (SYSTEM)

How many types of output records to other systems contain tags identifying the type of data they contain?

__/NA

CL.3.1.a (SYSTEM)

Has a common technical vocabulary with equivalent definitions been established for use with this system and for use with all interoperating systems? (For example, defining and using "DATA ITEM","BLOCK", "RECORD".)

Y/N/NA

CP.1.1.a (SYSTEM)

How many system capabilities have the inputs, processing, and outputs defined (documented with source, meaning, and format)?

__/NA

CP.1.2.a (SYSTEM)

How many data items are identified?

__/NA

CP.1.3.a (SYSTEM)

How many identified data items are defined (documented with regard to their source, meaning, and format)?

__/NA

CP.1.4.a (SYSTEM)

How many identified data items are defined, computed, or obtained from an external source? (For example, referencing: global data with preassigned values, input parameters with preassigned values.)

__/NA

CP.1.5.a (SYSTEM)

How many data items are referenced?

__/NA

CP.1.6.a (SYSTEM)

How many defined data items are referenced?

__/NA

CP.1.7.a (SYSTEM)

How many system capabilities have been allocated to configuration items (CSCIs, HWCIs)?

__/NA

CP.1.8.a (SYSTEM)

How many defined capabilities (documented with precise input, processing, output requirements) have been referenced?

__/NA

CP.1.9.a (SYSTEM)

How many referenced capabilities have been defined (documented with precise input, processing, output requirements)?

__/NA

CP.1.10.a (SYSTEM)

How many system capabilities have the flow of processing (algorithms) and all decision points (conditions and alternate paths) in that flow described?

__/NA

CP.1.12.a (SYSTEM)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.a (SYSTEM)

How many recorded problem reports have been corrected to date?

__/NA

CS.1.1.a (SYSTEM)

Is there a requirement to standardize all design representations? (For example, representing control flow, data flow.)

Y/N/NA

CS.1.2.a (SYSTEM)

Have specific standards been established for design representations? (For example, HIPO Charts, Program Design Language, Data Flow Diagrams.)

Y/N/NA

CS.1.4.a (SYSTEM)

Is there a requirement to standardize the calling sequence protocol between CSUs?

Y/N/NA

CS.1.5.a (SYSTEM)

Have specific standards been established for the calling sequence protocol between CSUs?

Y/N/NA

CS.1.7.a (SYSTEM)

Is there a requirement to standardize the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.8.a (SYSTEM)

Have specific standards been established for the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.10.a (SYSTEM)

Is there a requirement to standardize error propagation (passing) and handling (including rules for how exception handlers may be used) for all CSUs?

Y/N/NA

CS.1.11.a (SYSTEM)
Have specific standards been established for error handling for all CSUs?
Y/N/NA

CS.1.12.a (SYSTEM)
Do all references to the same system capability use single unique names?
Y/N/NA

CS.2.1.a (SYSTEM)
Is there a requirement to standardize references to data (e.g., as specified by the Ada package specifications) in the design?
Y/N/NA

CS.2.2.a (SYSTEM)
Have specific standards been established for data representation within the design?
Y/N/NA

CS.2.4.a (SYSTEM)
Is there a requirement to standardize naming conventions for all data?
Y/N/NA

CS.2.5.a (SYSTEM)
Have specific standards been established for naming conventions of all data?
Y/N/NA

CS.2.7.a (SYSTEM)
Is there a requirement to standardize the definition and use of all global variables?
Y/N/NA

CS.2.8.a (SYSTEM)
Have specific standards been established for defining and using all global variables?
Y/N/NA

CS.2.11.a (SYSTEM)
Are there requirements to establish consistency and concurrency for multiple copies of the same information? (For example, copies at different nodes.)
Y/N/NA

CS.2.13.a (SYSTEM)

Are there requirements to verify consistency and concurrency between multiple copies (e.g., copies at different nodes) of the same information?

Y/N/NA

CS.2.14.a (SYSTEM)

Do all references to the same data use single unique names?

Y/N/NA

DI.1.1.a (SYSTEM)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all software capabilities and interfaces in the system?

Y/N/NA

DI.1.2.a (SYSTEM)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all the different types of system information and the information flow within the system?

Y/N/NA

DI.1.3.a (SYSTEM)

Are there requirements to organize and distribute information within the system? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.a (SYSTEM)

Are there requirements for file/library accessibility from each node?

Y/N/NA

DI.1.5.a (SYSTEM)

Are there requirements to provide alternate processing sources within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.6.a (SYSTEM)

Are there requirements to distribute all mission-critical capabilities over redundant elements/nodes?

Y/N/NA

DI.1.8.a (SYSTEM)

Are there requirements to distribute control capabilities across different nodes/elements to ensure system operation under anomalous conditions?

Y/N/NA

DI.1.9.a (SYSTEM)

Are there requirements to implement system capabilities across several physical structures (i.e., function and physical structure are not necessarily the same)?

Y/N/NA

DI.1.10.a (SYSTEM)

Are there requirements regarding the number of nodes that can be removed (such that each node still maintains its capability to communicate with all remaining nodes)?

Y/N/NA

DO.1.1.a (SYSTEM)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.a (SYSTEM)

Is the system requirement documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.2.a (SYSTEM)

Does the requirements documentation depict control and data flow (e.g.,graphic portrayal with accompanying explanation, PDL)?

Y/N/NA

DO.2.3.a (SYSTEM)

Does each document contain an indexing scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs.)

Y/N/NA

DO.2.4.a (SYSTEM)

Does all the software documentation have separate volumes or separations within a single volume based on system capabilities?

Y/N/NA

DO.2.5.a (SYSTEM)

Does the system requirements documentation characterize the operational capabilities of the software? (For example, identify performance parameters and limitations.)

Y/N/NA

DO.2.6.a (SYSTEM)

Does the system requirements documentation include functional interfaces, functional processing, and functional algorithms for all identified system capabilities?

Y/N/NA

EC.1.1.a (SYSTEM)

Have performance requirements and limitations for system communication efficiency been specified for each system capability?

Y/N/NA

EP.1.1.a (SYSTEM)

Have performance requirements and limitations for processing efficiency been specified for each system capability? (For example, flow time for processes, execution time.)

Y/N/NA

EP.1.3.a (SYSTEM)

Is there a requirement to use an optimizing compiler, the pragma OPTIMIZE (with the argument TIME), or to code in assembly language to optimize processing efficiency?

Y/N/NA

EP.1.9.a (SYSTEM)

Is the memory management of the system free from requirements for overlays?

Y/N/NA

EP.2.1.a (SYSTEM)

Have performance requirements and limitations for storing data to efficiently process it, been specified for each system capability?

Y/N/NA

EP.2.2.a (SYSTEM)

Are there requirements to efficiently process stored information? (For example, rapidly update files, arrays, buffers, etc.)

Y/N/NA

EP.2.9.a (SYSTEM)

Does the source code language(s) enable variable initialization (at compile time) when the variable is declared? (Guaranteed value of 'Y' for Ada.)

Y/N/NA

ES.1.1.a (SYSTEM)

Have performance requirements and limitations for storing data to efficiently utilize primary and secondary storage been specified for each system capability (e.g., data packing, dynamic memory management)?

Y/N/NA

ES.1.2.a (SYSTEM)

Does the memory management of the system incorporate virtual storage?

Y/N/NA

ES.1.3.a (SYSTEM)

Does the memory management of the system incorporate dynamic reallocation of physical memory space during execution (dynamic memory management)?

Y/N/NA

ES.1.4.a (SYSTEM)

Is there a requirement to use an optimizing compiler, the pragma OPTIMIZE (with the argument SPACE), or to code in assembly language to optimize storage efficiency?

Y/N/NA

ES.1.11.a (SYSTEM)

Are there requirements to avoid redundant storage of files and libraries?

Y/N/NA

ES.1.12.a (SYSTEM)

Are there requirements for variable(s) to be read/written to/from a file to save memory space?

Y/N/NA

FO.1.3.a (SYSTEM)

How many duplicated capability sets exist between this system and interoperating systems (i.e., the same function is performed in this system and in interoperating systems)?

__/NA

FO.1.4.a (SYSTEM)

In how many instances of duplicated capability sets will duplicated capabilities be deleted (leaving responsibility for performing the capability with one system)?

__/NA

FO.1.5.a (SYSTEM)

How many of the duplicated capability sets require synchronization of the functions within the set?

__/NA

FO.1.6.a (SYSTEM)

How many of the duplicated capability sets require redundancy management techniques/logic to enable system interoperability?

__/NA

FS.2.1.a (SYSTEM)

Are there requirements to construct capabilities in such a way to facilitate their use in other similar system applications?

Y/N/NA

FS.2.3.a (SYSTEM)

How many system capabilities were reused from other system applications?

__/NA

FS.2.4.a (SYSTEM)

Are all inputs documented as to the specific use and limitations of the data?

Y/N/NA

FS.2.5.a (SYSTEM)

Are all input/output formats specified and documented?

Y/N/NA

FS.2.6.a (SYSTEM)

Are all outputs documented as to the specific use and interpretation of the data?

Y/N/NA

FS.3.1.a (SYSTEM)

Are there requirements to provide the user with options for computation and output? (For example, user selecting type of coordinate system, output media, and format.)

Y/N/NA

FS.3.2.a (SYSTEM)

Are there requirements to enable modification of the resources allocated to capabilities? (For example, changing the amount of memory work space for a capability.)

Y/N/NA

GE.2.1.a (SYSTEM)

Are there requirements that system components (e.g., CSUs) be organized to perform single processing capabilities?

Y/N/NA

GE.2.3.a (SYSTEM)

Are there requirements that the system be free from machine-dependent operations?

Y/N/NA

GE.2.4.a (SYSTEM)

Are there requirements that the volume of data handled by the system be changeable without effecting the executable code?

Y/N/NA

GE.2.5.a (SYSTEM)

Are there requirements that the range of data input values be changeable without effecting the executable code? (For example, no error tolerances or range tests.)

Y/N/NA

ID.1.1.a (SYSTEM)

Is there a requirement to use a standard subset of the implementation language(s) for the system? ("NA" for Ada; there are no subsets for Ada.)

Y/N/NA

ID.1.2.a (SYSTEM)

Has a standard subset of the implementation language(s) been established for coding the system? (Guaranteed "NA" for Ada; there are no established subsets for Ada.)

Y/N/NA

ID.2.1.a (SYSTEM)

Are the same version and dialect of the implementation language(s) supported on other machines? (Guaranteed "Y" for Ada if a validated Ada compiler is used.)

Y/N/NA

MO.1.1.a (SYSTEM)

Are there requirements to develop all software capabilities and software elements according to structured design techniques? (For example, top-down design, object-oriented design.)

Y/N/NA

MO.1.2.a (SYSTEM)

Are all system capabilities and CSCIs developed according to structured design techniques?

Y/N/NA

MO.2.1.a (SYSTEM)

Are there requirements regarding the relationships among software entities (types of coupling allowed among software capabilities, CSCIs, CSCs, CSUs)? (For example, requirements to minimize content, common, and external coupling among software entities.) (See Coupling in Glossary.)

Y/N/NA

MO.2.4.a (SYSTEM)

Are there requirements regarding the relationship between the elements within a software entity (cohesion value for software capabilities CSCIs, CSCs, CSUs)? (For example, all software entities are required to reflect an average cohesion value of .6 or greater.) (See Cohesion Value in Glossary.)

Y/N/NA

OP.1.1.a (SYSTEM)

Have the operating characteristics of the system been specified (i.e., the normal and alternate procedures and actions performed by the system)?

Y/N/NA

OP.1.2.a (SYSTEM)

Are all the errors specified, which are to be reported to the operator/user?

Y/N/NA

OP.1.3.a (SYSTEM)

Are the required operator/user responses specified for all reported errors?

Y/N/NA

OP.1.4.a (SYSTEM)

Are there requirements to provide the operator with the capability to interrupt operation, obtain operational status, save and enter data, and continue processing?

Y/N/NA

OP.1.5.a (SYSTEM)

How many operations/responses are performed by the operator for a typical mission/job?

__/NA

OP.1.6.a (SYSTEM)

Are there requirements to specify the procedures for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.a (SYSTEM)

Are there requirements to maintain a hard copy log of all operator interactions with the system?

Y/N/NA

OP.1.8.a (SYSTEM)

Are there requirements to provide simple and consistent operator messages requiring simple and consistent operator responses (i.e., minimize the number of operator messages and response formats, and use the same format types throughout the system)?

Y/N/NA

OP.1.10.a (SYSTEM)

Are there requirements to report all access violations to the operator?

Y/N/NA

OP.1.11.a (SYSTEM)

Are there requirements specifying the appropriate response (by the operator, system or both) for all access violations?

Y/N/NA

OP.1.12.a (SYSTEM)

Are there requirements to enable the operator/system to obtain specific system (or network) resource status information, and reallocate resources?

Y/N/NA

OP.1.13.a (SYSTEM)

Are there requirements to enable the operator/user to select different nodes for different types of processing, or for retrieval of different information?

Y/N/NA

OP.1.14.a (SYSTEM)

Are there requirements to enable the operator/user to have the capability to manipulate data, regardless of the data's location in the system?

Y/N/NA

OP.1.15.a (SYSTEM)

Are there requirements to make system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.1.a (SYSTEM)

What are the total number of user input parameters to system capabilities?

__/NA

OP.2.2.a (SYSTEM)

How many user input parameters have default values?

__/NA

OP.2.3.a (SYSTEM)

How many different input formats must the user be familiar with?

__/NA

OP.2.4.a (SYSTEM)

How many user input parameters enable the user to provide a description along with the values? (For example, user inputs: "TARGETS = 2.")

__/NA

OP.2.6.a (SYSTEM)

Are there requirements to allow the user to review and modify all input data prior to execution?

Y/N/NA

OP.2.7.a (SYSTEM)

Are there requirements to terminate all user-input data by explicitly defined logical end of input?

Y/N/NA

OP.2.8.a (SYSTEM)

Are there requirements to provide the user options for input media? (For example, terminal, tape drive, card reader.)

Y/N/NA

OP.2.9.a (SYSTEM)

Can the user choose among options for input media?

Y/N/NA

OP.3.1.a (SYSTEM)

Are there requirements to provide the user with output control? (For example, selecting: specific outputs, output media, output formats, amount of output.)

Y/N/NA

OP.3.2.a (SYSTEM)

Is there a requirement for all outputs to the user to have unique descriptive labels for identifying the data?

Y/N/NA

OP.3.3.a (SYSTEM)

Is there a requirement to provide all output to the user in user-oriented measurement units (e.g., pages, columns, rows, seconds, minutes, hours)?

Y/N/NA

OP.3.4.a (SYSTEM)

How many different formats are output tr the user? (For example, CRT display arrangements, printer outputs.)

__/NA

OP.3.5.a (SYSTEM)

Are all user outputs separated into logical groups to facilitate user examination?

Y/N/NA

OP.3.6.a (SYSTEM)

Are there requirements for all error messages to identify the nature of the error to the user?

Y/N/NA

OP.3.7.a (SYSTEM)

Are there requirements to provide the user with options for output media?

Y/N/NA

OP.3.8.a (SYSTEM)

Are there requirements to establish a standard user command language for network information and data access?

Y/N/NA

RE.1.1.a (SYSTEM)

Are there requirements to ensure communication paths to all remaining nodes/communication links in the event of a failure of one node/link?

Y/N/NA

RE.1.2.a (SYSTEM)

Are there requirements for maintaining the integrity of all data values following the occurrence of anomalous conditions?

Y/N/NA

RE.1.3.a (SYSTEM)

Are there requirements to enable all disconnected nodes to rejoin the network after recovery, such that the processing capabilities of the system are not interrupted?

Y/N/NA

RE.1.4.a (SYSTEM)

Are there requirements to replicate all critical data in the system at two or more distinct nodes?

Y/N/NA

SD.2.1.a (SYSTEM)

Has the specific standard been established that each CSU prologue contain the CSU's function, author, version number, version date, inputs, outputs, algorithms, assumptions and limitations?

Y/N/NA

SD.2.2.a (SYSTEM)

Has a standard been established for the identification and placement of comments in the CSU?

Y/N/NA

SD.3.7.a (SYSTEM)

Has a standard format for the structure of CSUs been established?

Y/N/NA

SI.1.1.a (SYSTEM)

Are there diagrams identifying all system capabilities in a structured fashion? (For example, top-down hierarchical.)

Y/N/NA

SI.1.9.a (SYSTEM)

Are there requirements for a programming standard?

Y/N/NA

SI.1.10.a (SYSTEM)

Has a programming standard been established?

Y/N/NA

SI.2.1.a (SYSTEM)

Are there requirements to use a structured language (e.g., Ada) or preprocessor to implement the software?

Y/N/NA

SS.1.1.a (SYSTEM)

Are there requirements to control user input/output access in the system? (For example, user access is limited by identification and password checking.)

Y/N/NA

SS.1.2.a (SYSTEM)

Are there requirements to control data access in the system?

Y/N/NA

SS.1.3.a (SYSTEM)

Are there requirements to control the scope of task operations during execution? (For example, tasks cannot invoke other tasks, access system registers, or use privileged commands.)

Y/N/NA

SS.1.4.a (SYSTEM)

Are there requirements to control access to the network in the system?

Y/N/NA

SS.2.1.a (SYSTEM)

Are there requirements to record and report all access to the system? (For example, terminal and processor linkage, data file access, and job run information are recorded.)

Y/N/NA

SS.2.2.a (SYSTEM)

Are there requirements to immediately indicate and identify all access violations?

Y/N/NA

ST.3.1.a (SYSTEM)

Are there requirements to isolate I/O capabilities from computational capabilities?

Y/N/NA

ST.3.2.a (SYSTEM)

Are I/O capabilities isolated from computational capabilities?

Y/N/NA

SY.1.1.a (SYSTEM)

Are there requirements for this system's I/O transmission rate to be the same as the interoperating system(s)?

Y/N/NA

SY.1.2.a (SYSTEM)

Are there requirements for this system to use the same communication protocol as the interoperating system(s)?

Y/N/NA

SY.1.3.a (SYSTEM)

Are there requirements for common interpretation of the content in all messages sent from and received by this system and the interoperating system(s)? (For example, all variables in the message have the same meaning.)

Y/N/NA

SY.1.4.a (SYSTEM)

Are there requirements for this system to use the same structure and sequence for message contents, as the interoperating system(s)? (For example, all real variables are 16 bits in length, and real coordinates are ordered Xcoord, Ycoord, Zcoord.)

Y/N/NA

SY.2.1.a (SYSTEM)

Are there requirements for this system to use the same data format as the interoperating system(s)? (For example, all characters are represented in ASCII format.)

Y/N/NA

SY.2.2.a (SYSTEM)

Are there requirements for this system to establish the same data base structure as the interoperating system(s)? (For example, all systems use a relational data base containing similar information.)

Y/N/NA

SY.2.3.a (SYSTEM)

Are there requirements for this system to provide the same data base access techniques as the interoperating system(s)?

Y/N/NA

SY.3.1.a (SYSTEM)

Are there requirements for this system to use the same word length as the interoperating system(s)?

Y/N/NA

SY.3.2.a (SYSTEM)

Are there requirements for this system to use the same interrupt structure as the interoperating system(s)?

Y/N/NA

SY.3.3.a (SYSTEM)

Are there requirements for this system to use the same instruction set as the interoperating system(s)?

Y/N/NA

SY.4.1.a (SYSTEM)

Are there requirements for this system to use the same sòurce code language(s) as the interoperating system(s)?

Y/N/NA

SY.4.2.a (SYSTEM)

Are there requirements for this system to use the same operating system as the interoperating system(s)?

Y/N/NA

SY.4.3.a (SYSTEM)

Are there requirements for this system to use the same support software as the interoperating system(s)?

Y/N/NA

SY.5.1.a (SYSTEM)

Is documentation available from the interoperating system(s), that enables interoperability requirements to be established for this system? (For example, documentation is up-to-date, complete, and clearly organized.)

Y/N/NA

TN.1.1.a (SYSTEM

Are there requirements to provide lesson plans and training materials for operators, end users, and maintainers of the system?

Y/N/NA

TN.1.2.a (SYSTEM)

Are there requirements to provide realistic simulation exercises for the system?

Y/N/NA

TN.1.3.a (SYSTEM)

Are there requirements to provide "HELP" information and diagnostic information to the operator, end user and maintainer of the system? (For example, an on-line list of legal commands or a list of the sequential steps in a process are provided.)

Y/N/NA

TN.1.4.a (SYSTEM)

Are there requirements to provide selectable levels of aid and guidance for system users of different degrees of expertise?

Y/N/NA

VS.1.4.a (SYSTEM)

Is there a requirement to test all input parameters?

Y/N/NA

VS.2.2.a (SYSTEM)

Is there a requirement to test all interfaces?

Y/N/NA

VS.3.1.a (SYSTEM)

Are there requirements that specified performance requirements be tested?

Y/N/NA

VS.3.2.a (SYSTEM)

Are there requirements that all components of the system be exercised during testing?

Y/N/NA

VS.3.3.a (SYSTEM)

Is there a requirement for a summary table listing inputs and outputs for testing?

Y/N/NA

VS.4.1.a (SYSTEM)

Is there a requirement to test specified system capabilities?

Y/N/NA

DATA COLLECTION FORM -- B-LEVEL

AC.1.4.b (CSCI)

Are there quantitative accuracy requirements for all applicable inputs associated with each mission critical capability?

Y/N/NA

AC.1.5.b (CSCI)

Are there quantitative accuracy requirements for all applicable outputs associated with each mission critical capability?

Y/N/NA

AC.1.6.b (CSCI)

Are there quantitative accuracy requirements for all applicable constants associated with each mission critical capability?

Y/N/NA

AC.1.7.b (CSCI)

Do the existing math library subprograms which are planned for use in the CSCI provide enough precision to support accuracy objectives?

Y/N/NA

AM.1.1.b (CSCI)

How many instances are there of different processes (capabilities, constituent capabilities) which are required to execute at the same time (concurrent processing, tasking)?

__/NA

AM.1.2.b (CSCI)

In how many instances is concurrent processing or tasking centrally controlled?

__/NA

AM.1.3.b (CSCI)

How many error conditions are identified?

__/NA

AM.1.4.b (CSCI)

How many recognized error conditions require recovery or repair of the error?

__/NA

AM.4.1.b (CSCI)

Are there requirements for recovery (i.e., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, power failure, clock interrupt)?

Y/N/NA

AM.5.1.b (CSCI)

Are there requirements for recovery from all I/O device errors (i.e., I/O Exception - Device Error)?

Y/N/NA

AM.6.1.b (CSCI)

Are there requirements to recover (i.e., exception handlers or other means) from all communication transmission errors?

Y/N/NA

AM.6.2.b (CSCI)

Are there requirements to transmit messages with error checking information (for example, check sums, parity bit)?

Y/N/NA

AM.6.3.b (CSCI)

Are there requirements to compute and compare error checking information upon receipt of all messages?

Y/N/NA

AM.6.4.b (CSCI)

Are there requirements to limit retries for all messages?

Y/N/NA

AM.7.1.b (CSCI)

Are there requirements to recover from all failures to communicate with other nodes/systems?

Y/N/NA

AM.7.2.b (CSCI)

Is there a requirement to periodically check adjacent nodes or interoperating systems for operational status?

Y/N/NA

AM.7.3.b (CSCI)

Is there a requirement to provide a strategy for alternate routing of messages?

Y/N/NA

AP.1.1.b (CSCI)

Is there a requirement to limit specific references to the database management scheme? (For example, all data calls to database information are processed through an executive.)

Y/N/NA

AP.2.3.b (CSCI)

Is there a standard for commenting all global data within a CSU to show where the data is derived, the data's composition, and how the data is used?

Y/N/NA

AP.2.4.b (CSCI)

Is there a standard for commenting all parameter input/output and local variables in CSUs which includes the data's composition and use?

Y/N/NA

AP.3.1.b (CSCI)

Is there a requirement to localize specific references to computer architecture (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.b (CSCI)

Is there a requirement to avoid or limit the use of microcode instruction statements in the CSCI?

Y/N/NA

AP.5.1.b (CSCI)

Is there a requirement to develop functional processing algorithms which are not unique to this CSCI's application (i.e., can be used in similar applications with minimal tailoring)

Y/N/NA

AP.5.2.b (CSCI)

How many CSCI capabilities are there?

__/NA

AP.5.3.b (CSCI)

How many capabilities were reused from other applications?

__/NA

AP.5.6.b (SYSTEM)

How many CSCIs are there?

__/NA

AT.1.2.b (CSCI)

Are there requirements for spare memory storage capacity?

Y/N/NA

AT.1.5.b (CSCI)

Are there requirements for spare auxiliary storage capacity?

Y/N/NA

AT.2.3.b (CSCI)

Are there requirements for spare processing capacity (time)?

Y/N/NA

AT.3.1.b (CSCI)

Are there requirements for spare I/O channel capacity (time)?

Y/N/NA

AT.3.4.b (CSCI)

Are there requirements for spare communication channel capacity (time)?

Y/N/NA

AT.4.1.b (CSCI)

Are there requirements for interface compatibility among all processors, communication links, memory devices, and peripherals within the CSCI?

Y/N/NA

AT.4.2.b (CSCI)

Is documentation available which describes the results of previous engineering studies (such as trade-off studies, feasibility studies, risk analysis, requirement definitions)?

Y/N/NA

AT.4.3.b (CSCI)

Is documentation available which describes new or emerging software areas which may affect the scope of the software requirements or the software implementation techniques? (For example, voice recognition using AI techniques.)

Y/N/NA

AU.2.2.b (CSCI)

Are there requirements for executive software to test its own operation, communication links, memory devices, and peripheral devices?

Y/N/NA

CL.1.1.b (CSCI)

Are there requirements for communication with other systems?

Y/N/NA

CL.1.2.b (CSCI)

Is there a requirement for protocol standards to control all network communications?

Y/N/NA

CL.1.3.b (CSCI)

How many CSCI capabilities receive input from other CSCIs?

__/NA

CL.1.4.b (CSCI)

How many CSCI capabilities transmit output to other system?

__/NA

CL.1.5.b (CSCI)

Are there requirements to control all network processing?

Y/N/NA

CL.1.6.b (CSCI)

Is network processing control part of the protocol standards?

Y/N/NA

CL.1.7.b (CSCI)

Are there requirements to control all user sessions?

Y/N/NA

CL.1.8.b (CSCI)

Is user session control part of the network protocol standards?

Y/N/NA

CL.1.9.b (CSCI)

Are there requirements for a communications routing strategy?

Y/N/NA

CL.1.10.b (CSCI)
Is a communication routing strategy part of the network protocol standards?

Y/N/NA

CL.1.11.b (CSCI)
Are there requirements to handle messages (e.g., synchronization, message decoding) in a uniform manner?

Y/N/NA

CL.1.12.b (CSCI)
Is uniform message handling (e.g., synchronization, message decoding) part of the network protocol standards?

Y/N/NA

CL.1.13.b (CSCI)
How many other systems must respond correctly to successfully complete synchronization?

__/NA

CL.1.14.b (CSCI)
Is the system free from synchronization imposing constraints upon CSCI computation or response time (e.g., resulting in a user wait time of more than a few seconds)?

Y/N/NA

CL.1.15.b (CSCI)
Is the CSCI free from time-critical performance requirements for system communication? (For example, data freshness.)

Y/N/NA

CL.1.16.b (CSCI)
How many other systems is this CSCI required to interface with?

__/NA

CL.1.17.b (CSCI)
Is there a description of how the computer system will appear to its users and how the users will interact with the system?

Y/N/NA

CL.1.18.b (CSCI)
Is there a complete and definitive set of operating procedures for using this system?

Y/N/NA

CL.2.1.b (CSCI)

Are there requirements for a standard to establish common representations of data for uniform communications with other systems?

Y/N/NA

CL.2.2.b (CSCI)

Has a standard been established for common representations of data and/or translations between representations of data for uniform communication with other systems?

Y/N/NA

CL.2.3.b (CSCI)

How many capabilities perform data translations?

__/NA

CL.2.4.b (CSCI)

Is there a requirement to receive all input data from other systems in common formats? (For example, common format for data positioning, data packing, block transmission.)

Y/N/NA

CL.2.5.b (CSCI)

How many different formats are used for input data from other systems (formats for data positioning, data packing, block transmission)?

__/NA

CL.2.6.b (CSCI)

Is there a requirement to output all data to other systems in common formats (common formats for data positioning, data packing, block transmission)?

Y/N/NA

CL.2.7.b (CSCI)

How many different formats are used to output data to other systems (e.g.,formats for data positioning, data packing, block transmission)?

__/NA

CL.2.8.b (CSCI)

How many different types of input records are received from other systems?

__/NA

CL.2.9.b (CSCI)

How many types of input records transmitted from other systems contain tags identifying the type of data they contain?

__/NA

CL.2.10.b (CSCI)

How many different types of output records are transmitted to other systems?

__/NA

CL.2.11.b (CSCI)

How many types of output records to other systems contain tags identifying the type of data they contain?

__/NA

CL.3.1.b (CSCI)

Has a common technical vocabulary with equivalent definitions been established for use with this system and for use with all interoperating system? (For example, defining and using "DATA ITEM", "BLOCK", "RECORD".)

Y/N/NA

CP.1.1.b (CSCI)

How many identified capabilities have inputs, processing, and outputs all defined (documented with source, meaning and format)?

__/NA

CP.1.2.b (CSCI)

How many data items are identified?

__/NA

CP.1.3.b (CSCI)

How many identified data items are defined (documented with regard to their source, meaning, and format)?

__/NA

CP.1.4.b (CSCI)

How many identified data items are defined, computed, or obtained from an external source? (For example, referencing: global data with preassigned values, input parameters with preassigned values.)

__/NA

CP.1.5.b (CSCI)

How many data items are referenced?

__/NA

CP.1.6.b (CSCI)

How many defined data items are referenced?

__/NA

CP.1.7.b (CSCI)

Have all system capabilities allocated to this CSCI been allocated to software capabilities within this CSCI?

Y/N/NA

CP.1.8.b (CSCI)

How many defined capabilities (documented with precise input, processing, output requirements) have been referenced?

__/NA

CP.1.9.b (CSCI)

How many referenced capabilities have been defined (documented with precise input, processing, output requirements)?

__/NA

CP.1.10.b (CSCI)

How many CSCI capabilities have the flow of processing (algorithms) and all decision points (conditions and alternate paths) in that flow described?

__/NA

CP.1.12.b (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.b (CSCI)

How many recorded problem reports have been corrected to date?

__/NA

CS.1.1.b (CSCI)

Is there a requirement to standardize all design representations? (For example, representing control flow, data flow.)

Y/N/NA

CS.1.2.b (CSCI)

Have specific standards been established for design representations? (For example, HIPO Charts, Program Design Language, Data Flow Diagrams.)

Y/N/NA

CS.1.4.b (CSCI)

Is there a requirement to standardize the calling sequence protocol between CSUs?

Y/N/NA

CS.1.5.b (CSCI)

Have specific standards been established for the calling sequence protocol between CSUs?

Y/N/NA

CS.1.7.b (CSCI)

Is there a requirement to standardize the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.8.b (CSCI)

Have specific standards been established for the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.10.b (CSCI)

Is there a requirement to standardize error propagation (passing) and handling (including rules for how exception handlers may be used) for all CSUs?

Y/N/NA

CS.1.11.b (CSCI)

Have specific standards been established for error handling for all CSUs?

Y/N/NA

CS.1.12.b (CSCI)

For how many CSCI capabilities do all references to that capability use a single, unique name?

__/NA

CS.2.1.b (CSCI)

Have specific standards been established for all references to data (e.g., as specified by the Ada package specifications) in the design?

Y/N/NA

CS.2.2.b (CSCI)

Have specific standards been established for data representation within the design?

Y/N/NA

CS.2.4.b (CSCI)

Is there a requirement to standardize naming conventions for all data?

Y/N/NA

CS.2.5.b (CSCI)

Have specific standards been established for naming conventions of all data?

Y/N/NA

CS.2.7.b (CSCI)

Is there a requirement to standardize the definition and use of all global variables?

Y/N/NA

CS.2.8.b (CSCI)

Have specific standards been established for defining and using all global variables?

Y/N/NA

CS.2.11.b (CSCI)

Are there procedures to establish consistency and concurrency for multiple copies of the same information? (For example, copies at different nodes.)

Y/N/NA

CS.2.13.b (CSCI)

Are there procedures to verify consistency and concurrency between multiple copies (e.g., copies at different nodes) of the same information?

Y/N/NA

CS.2.14.b (CSCI)

For how many defined data items do all references to that item use a single, unique name?

__/NA

DI.1.1.b (CSCI)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all software capabilities and functional interfaces?

Y/N/NA

DI.1.2.b (CSCI)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all the different types of CSCI information and the information flow within the CSCI?

Y/N/NA

DI.1.3.b (CSCI)

Are there requiremen.s to organize and distribute information within the CSCI? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.b (CSCI)

Are there requirements for file/library accessibility from each node?

Y/N/NA

DI.1.6.b (CSCI)

Are there requirements to distribute all mission-critical capabilities over redundant elements/nodes?

Y/N/NA

DI.1.8.b (CSCI)

Are there requirements to distribute control capabilities across different nodes/elements to ensure system operation under anomalous conditions?

Y/N/NA

DI.1.9.b (CSCI)

Do the requirements allow for implementing capabilities across several physical structures (i.e., function and physical structures are not necessarily the same)?

Y/N/NA

DI.1.10.b (CSCI)

Are there requirements regarding the number of nodes that can be removed (such that each node still maintains its capability to communicate with all remaining nodes)?

Y/N/NA

DO.1.1.b (CSCI)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.b (CSCI)

Is the documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.2.b (CSCI)

Does the requirements/design documentation depict control and data flow (e.g., graphic portrayal with accompanying explanation, PDL)?

Y/N/NA

DO.2.3.b (CSCI)

Does each document contain an indexing scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs.)

Y/N/NA

DO.2.4.b (CSCI)

Does all the software documentation have separate volumes or separations within a single volume based on software capabilities?

Y/N/NA

DO.2.5.b (CSCI)

Does the documentation characterize the operational capabilities of the software? (For example, identify performance parameters and limitations.)

Y/N/NA

DO.2.6.b (CSCI)

Does the documentation include functional interfaces, function processing, and function algorithms for all identified CSCI capabilities?

Y/N/NA

EC.1.1.b (CSCI)

Have performance requirements and limitations for system communication efficiency been specified for each CSCI capability?

Y/N/NA

EP.1.1.b (CSCI)

Have performance requirements and limitations for processing efficiency been specified for each CSCI capability? (For example, flow time for processes, execution time.)

Y/N/NA

EP.1.3.b (CSCI)

Is there a requirement to use an optimizing compiler, the pragma OPTIMIZE (with the argument TIME), or to code in assembly language to optimize processing efficiency?

Y/N/NA

EP.1.9.b (CSCI)

Is the memory management of the CSCI free from requirements for overlays?

Y/N/NA

EP.2.1.b (CSCI)

Have performance requirements and limitations for storing data to efficiently process it, been specified for each CSCI capability?

Y/N/NA

EP.2.2.b (CSCI)

Are there requirements to efficiently process stored information? (For example, rapidly update files, arrays, buffers, etc.)

Y/N/NA

EP.2.9.b (CSCI)

Does the source code language(s) enable variable initialization (at compile time) when the variable is declared? (Guaranteed value of "Y" for Ada.)

Y/N/NA

ES.1.1.b (CSCI)

Have performance requirements and limitations for storing data to efficiently utilize primary and secondary storage been specified for each CSCI capability (e.g., data packing, dynamic memory management)?

Y/N/NA

ES.1.2.b (CSCI)

Does the memory management of the CSCI include virtual storage?

Y/N/NA

ES.1.3.b (CSCI)

Does the memory management of the CSCI incorporate dynamic reallocation of physical memory space during execution (dynamic memory management)?

Y/N/NA

ES.1.4.b (CSCI)

Is there a requirement to use an optimizing compiler, the pragma OPTIMIZE (with the argument SPACE), or to code in assembly language, to optimize storage efficiency?

Y/N/NA

ES.1.11.b (CSCI)

Are there requirements to avoid redundant storage of files and libraries?

Y/N/NA

ES.1.12.b (CSCI)

Are there requirements for a variable(s) to be read/written to/from a file in order to save memory?

Y/N/NA

FO.1.3.b (CSCI)

How many duplicated capability sets exist between this CSCI and interoperating systems (i.e.,the same capability is performed in this CSCI and in interoperating systems)?

__/NA

FO.1.4.b (CSCI)

In how many instances of duplicated capability sets will duplicated capabilities be deleted (leaving responsibility for performing the function with one system)?

__/NA

FO.1.5.b (CSCI)

How many of the duplicated capability sets require synchronization of the capabilities within the set?

__/NA

FO.1.6.b (CSCI)

How many of the duplicated capability sets require redundant management techniques/logic to enable system interoperability?

__/NA

FS.2.1.b (CSCI)

Are there requirements to construct capabilities in such a way to facilitate their use in other similar CSCI applications?

Y/N/NA

FS.2.3.b (CSCI)

How many system capabilities were reused from other system applications?

__/NA

FS.2.4.b (CSCI)

Are all inputs documented as to the specific use and limitations of the data?

Y/N/NA

FS.2.5.b (CSCI)

Are all input/output formats specified and documented?

Y/N/NA

FS.2.6.b (CSCI)

Are all outputs documented as to the specific use and interpretation of the data?

Y/N/NA

FS.3.1.b (CSCI)

Are there requirements to provide the user options for computation and output? (For example, user selecting type of coordinate system, output media, and format.)

Y/N/NA

FS.3.2.b (CSCI)

Are there requirements to enable modification of the resources allocated to capabilities? (For example, changing the amount of memory work space for a capability.)

Y/N/NA

GE.2.1.b (CSCI)

Are there requirements that system components (e.g., CSUs) be organized to perform single processing capabilities?

Y/N/NA

GE.2.3.b (CSCI)

Are there requirements that the system be free from machine-dependent operations?

Y/N/NA

GE.2.4.b (CSCI)

Are there requirements that the volume of data handled by the system be changeable without effecting the executable code?

Y/N/NA

GE.2.5.b (CSCI)

Are there requirements that the range of data input values be changeable without effecting the executable code? (For example, no error tolerances or range tests.)

Y/N/NA

ID.1.1.b (CSCI)

Is there a requirement to use a standard subset of the implementation language(s) for the CSCI? ("NA" for Ada; there are not subsets for Ada.)

Y/N/NA

ID.1.2.b (CSCI)

Has a standard subset of the implementation language(s) been established for coding the CSCI? (Guaranteed "NA" for Ada; there are no established subsets for Ada.)

Y/N/NA

ID.2.1.b (CSCI)

Are the same version and dialect of the implementation language(s) supported on other machines? (Guaranteed "Y" for Ada if a validated compiler is used.)

Y/N/NA

MO.1.1.b (CSCI)

Are there requirements to develop all software capabilities and software elements according to structured design techniques? (For example, top-down design, object-oriented design.)

Y/N/NA

MO.1.2.b (CSCI)

Are all software capabilities and CSCs developed according to structured design techniques?

Y/N/NA

MO.2.1.b (CSCI)

Are there requirements regarding the relationships among software entities (types of coupling allowed among software capabilities, CSCIs, CSCs,CSUs)? (For example, requirements to minimize content, common, and external coupling among software entities.) (See Coupling in Glossary.)

Y/N/NA

MO.2.2.b (CSCI)

How many interfaces among software capabilities in this CSCI?

__/NA

MO.2.3.b1 (CSCI)

How many software interfaces include content coupling?

__/NA

MO.2.3.b2 (CSCI)

How many software interfaces include common coupling?

__/NA

MO.2.3.b3 (CSCI)

How many software interfaces include external coupling?

__/NA

MO.2.3.b4 (CSCI)

How many software interfaces include control coupling?

__/NA

MO.2.3.b5 (CSCI)

How many software interfaces include stamp coupling?

__/NA

MO.2.3.b6 (CSCI)

How many software interfaces include data coupling?

__/NA

MO.2.4.b (CSCI)

Are there requirements regarding the relationship between the elements within a software entity (cohesion value for software capabilities, CSCIs, CSCs, CSUs)? (For example, all software entities are required to reflect an average cohesion value of .6 or greater.) (See Cohesion Value in Glossary.)

Y/N/NA

MO.2.5.b (CSCI)

What is the cohesion value of the CSCI? (Functional = 1.0, Informational = 0.7, Communicational = 0.5, Procedural = 0.3, Classical = 0.1, Logical = 0.1, Coincidental = 0.0)

__/NA

OP.1.1.b (CSCI)

Have the operating characteristics of the CSCI been specified (i.e., the normal and alternate procedures and actions performed by the CSCI)?

Y/N/NA

OP.1.2.b (CSCI)

Are all the errors specified which are to be reported to the operator/user?

Y/N/NA

OP.1.3.b (CSCI)

Are the required operator/user responses specified for all reported errors?

Y/N/NA

OP.1.4.b (CSCI)

Are there requirements to provide the operator with the capability to interrupt operation, obtain operational status, save and enter data, and continue processing?

Y/N/NA

OP.1.5.b (CSCI)

How many operations/responses are performed by the operator for a typical mission/job?

__/NA

OP.1.6.b (CSCI)

Are there requirements to specify the procedures for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.b (CSCI)

Are there requirements to maintain a hard copy log of all operator interactions with the CSCI?

Y/N/NA

OP.1.8.b (CSCI)

Are there requirements to provide simple and consistent operator messages, and require simple and consistent operator responses (i.e., minimize the number of operator messages and response formats, and use the same format types throughout the CSCI)?

Y/N/NA

OP.1.10.b (CSCI)

Are there requirements to report all access violations to the operator?

Y/N/NA

OP.1.11.b (CSCI)

Are there requirements specifying the appropriate response (by the operator, CSCI, or both) for all access violations?

Y/N/NA

OP.1.12.b (CSCI)

Are there requirements to provide the operator/software the capability to obtain specific system (or network) resource status information, and reallocate resources?

Y/N/NA

OP.1.13.b (CSCI)

Are there requirements to provide the operator/user the capability to select different nodes for different types of processing, or for retrieval of different information?

Y/N/NA

OP.1.14.b (CSCI)

Are there requirements to provide the operator/user to have the capability to manipulate data, regardless of the data's location in the system?

Y/N/NA

OP.1.15.b (CSCI)

Are there requirements to make system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.1.b (CSCI)

What are the total number of user input parameters to CSCI capabilities?

__/NA

OP.2.2.b (CSCI)

How many user input parameters have default values?

__/NA

OP.2.3.b (CSCI)

How many different input formats must the user be familiar with?

__/NA

OP.2.4.b (CSCI)

How many user input parameters enable the user to provide a description along with the values? (For example, user inputs: "TARGETS = 2.")

__/NA

OP.2.6.b (CSCI)

Are there requirements to allow the user to review and modify all input data prior to execution?

Y/N/NA

OP.2.7.b (CSCI)

Are there requirements to terminate all user-input data by explicitly defined logical end of input?

Y/N/NA

OP.2.8.b (CSCI)

Are there requirements to provide the user options for input media? (For example, terminal, tape drive, card reader.)

Y/N/NA

OP.3.1.b (CSCI)

Are there requirements to provide the user with output control? (For example, selecting: specific outputs, output media, output formats, amount of output.)

Y/N/NA

OP.3.2.b (CSCI)

Is there a requirement for all outputs to the user to have unique descriptive labels for identifying the data?

Y/N/NA

OP.3.3.b (CSCI)

Is there a requirement to provide all output to the user in user-oriented measurement units (e.g., pages, columns, rows, seconds, minutes, hours)?

Y/N/NA

OP.3.4.b (CSCI)

How many different formats are output to the user? (For example, CRT display arrangements, printer outputs.)

__/NA

OP.3.5.b (CSCI)

Are all user outputs separated into logical groups to facilitate user examination?

Y/N/NA

OP.3.6.b (CSCI)

Are there requirements for all error messages to identify the nature of the error to the user?

Y/N/NA

OP.3.7.b (CSCI)

Are there requirements to provide the user with options for output media?

Y/N/NA

OP.3.8.b (CSCI)

Are there requirements to establish a standard user command language for network information and data access?

Y/N/NA

RE.1.1.b (CSCI)

Are there requirements to ensure communication paths to all remaining nodes/communication links in the event of a failure of one node/link?

Y/N/NA

RE.1.2.b (CSCI)

Are there requirements for maintaining the integrity of all data values following the occurrence of anomalous conditions?

Y/N/NA

RE.1.3.b (CSCI)

Are there requirements to enable all disconnected nodes to rejoin the network after recovery, such that the processing capabilities of the CSCI are not interrupted?

Y/N/NA

RE.1.4.b (CSCI)

Are there requirements to replicate all critical data in the CSCI at two or more distinct nodes?

Y/N/NA

SD.2.1.b (CSCI)

Has the specific standard been established that each CSU prologue contain the CSU's function, author, version number, version date, inputs, outputs, algorithms, assumptions and limitations?

Y/N/NA

SD.2.2.b (CSCI)

Has a standard been established for the identification and placement of comments in the CSU?

Y/N/NA

SD.3.7.b (CSCI)

Has a standard format for the structure of CSUs been established?

Y/N/NA

SI.1.1.b (CSCI)

Are there diagrams identifying all capabilities in a structured fashion? (For example, top-down hierarchical.)

Y/N/NA

SI.1.9.b (CSCI)

Are there requirements for a programming standard?

Y/N/NA

SI.1.10.b (CSCI)

Has a programming standard been established?

Y/N/NA

SI.2.1.b (CSCI)

Are there requirements to use a structured language (e.g., Ada) or preprocessor to implement the software?

Y/N/NA

SS.1.1.b (CSCI)

Are there requirements to control user input/output access in the CSCI? (For example, user access is limited by identification and password checking.)

Y/N/NA

SS.1.2.b (CSCI)

Are there requirements to control data access in the CSCI?

Y/N/NA

SS.1.3.b (CSCI)

Are there requirements to control the scope of task operations during execution? (For example, tasks cannot invoke other tasks, access system registers, or use privileged commands.)

Y/N/NA

SS.1.4.b (CSCI)

Are there requirements to control access to the network in the CSCI?

Y/N/NA

SS.2.1.b (CSCI)

Are there requirements to record and report all access to the system? (For example, terminal and processor linkage, data file access, and job run information are recorded.)

Y/N/NA

SS.2.2.b (CSCI)

Are there requirements to immediately indicate and identify all access violations?

Y/N/NA

ST.3.1.b (CSCI)

Are there requirements to isolate I/O capabilities from computational capabilities?

Y/N/NA

ST.3.2.b (CSCI)

Are I/O capabilities isolated from computational capabilities?

Y/N/NA

SY.1.1.b (CSCI)

Are there requirements for this CSCI's I/O transmission rate to be the same as the interoperating system(s)?

Y/N/NA

SY.1.2.b (CSCI)

Are there requirements for this CSCI to use the same communication protocol as the interoperating system(s)?

Y/N/NA

SY.1.3.b (CSCI)

Are there requirements for common interpretation of the content in all messages sent from and received by this CSCI and the interoperating system(s)? (For example, all variables in the message have the same meaning.)

Y/N/NA

SY.1.4.b (CSCI)

Are there requirements for this CSCI to use the same structure and sequence for message contents, as the interoperating system(s)? (For example, all real variables are 16 bits in length, and real coordinates are ordered Xcoord, Ycoord, Zcoord.)

Y/N/NA

SY.2.1.b (CSCI)

Are there requirements for this CSCI to use the same data format as the interoperating system(s)? (For example, all characters are represented in ASCII format.)

Y/N/NA

SY.2.2.b (CSCI)

Are there requirements for this CSCI to establish the same data base structure as the interoperating system(s)? (For example, all systems use a relational data base containing similar information.)

Y/N/NA

SY.2.3.b (CSCI)

Are there requirements for this CSCI to provide the same data base access techniques as the interoperating system(s)?

Y/N/NA

SY.3.1.b (CSCI)

Are there requirements for this CSCI to use the same word length as the interoperating system(s)?

Y/N/NA

SY.3.2.b (CSCI)

Are there requirements for this CSCI to use the same interrupt structure as the interoperating system(s)?

Y/N/NA

SY.3.3.b (CSCI)

Are there requirements for this CSCI to use the same instruction set as the interoperating system(s)?

Y/N/NA

SY.4.1.b (CSCI)

Are there requirements for this CSCI to use the same source code language(s) as the interoperating system(s)?

Y/N/NA

SY.4.2.b (CSCI)

Are there requirements for this CSCI to use the same operating system as the interoperating system(s)?

Y/N/NA

SY.4.3.b (CSCI)

Are there requirements for this CSCI to use the same support software as the interoperating system(s)?

Y/N/NA

SY.5.1.b (CSCI)

Is documentation from the interoperating system(s) available that enables interoperability requirements to be established for this CSCI? (For example, documentaticn is up-to-date, complete, and clearly organized.)

Y/N/NA

TC.1.1.b (CSCI)

Is there a table(s) tracing all of the CSCI's allocated requirements to the parent system or subsystem specification(s)?

Y/N/NA

TN.1.1.b (CSCI)

Are there requirements to provide lesson plans and training materials for operators, end users, and maintainers of the CSCI?

Y/N/NA

TN.1.2.b (CSCI)

Are there requirements to provide realistic simulation exercises for the CSCI?

Y/N/NA

TN.1.3.b (CSCI)

Are there requirements to provide "HELP" information and diagnostic information to the operator, end user and maintainer of the CSCI? (For example, an on-line list of legal commands or a list of the sequential steps in a process, are provided.)

Y/N/NA

TN.1.4.b (CSCI)

Are there requirements to provide selectable levels of aid and guidance for CSCI users of different degrees of expertise?

Y/N/NA

VS.1.4.b (CSCI)

Is there a requirement to test all input parameters?

Y/N/NA

VS.2.2.b (CSCI)

Is there a requirement to test all interfaces?

Y/N/NA

VS.3.1.b (CSCI)

Are there requirements that specified performance requirements be tested?

Y/N/NA

VS.3.2.b (CSCI)

Are there requirements that all CSCs be exercised during testing?

Y/N/NA

VS.3.3.b (CSCI)

Is there a requirement for a summary table listing inputs and outputs for testing?

Y/N/NA

VS.4.1.b (CSCI)

Is there a requirement to test specified CSCI capabilities?

Y/N/NA

DATA COLLECTION FORM -- C-LEVEL

AC.1.3.c (CSC)

Do the numerical techniques used in the CSC implementing mission-critical capabilities provide enough precision to support accuracy objectives?

Y/N/NA

AC.1.4.c (CSC)

Are there qualitative accuracy requirements for all applicable inputs associated with each mission critical capability?

Y/N/NA

AC.1.5.c (CSC)

Are there quantitative accuracy requirements for all applicable outputs associated with each mission critical capability?

Y/N/NA

AC.1.6.c (CSC)

Are there quantitative accuracy requirements for all applicable constants associated with each mission critical capability?

Y/N/NA

AC.1.7.c (CSC)

Do the existing math library subprograms which are planned for use in the CSC provide enough precision too support accuracy objectives

Y/N/NA

AM.1.3.c (CSC)

How many error conditions are identified?

__/NA

AM.1.4.c (CSC)

How many identified error conditions are provided with processing instructions for recovery or repair of the error?

__/NA

AM.1.5.c (CSC)

Is there a standard for handling recognized errors such that all error conditions are reported (via raising, propagating exceptions, or passing a value) to the calling body (e.g., subprogram, task, or package)?

Y/N/NA

AM.1.6.c (CSC)

How many instances are there of the same process (capability, constituent capability) being required to execute more than once for comparison purposes? (For example, polling parallel processing, redundant processing, or tasking results.)

__/NA

AM.1.7.c (CSC)

How many instances of redundant processing are required to be centrally controlled?

__/NA

AM.2.1.c (CSCI)

Are error tolerances specified for all particular external input data (e.g., range of numerical values, legal combinations of alphanumeric values) to the CSCI?

Y/N/NA

AM.2.2.c (CSC)

Are values of all applicable inputs range-specified?

Y/N/NA

AM.2.3.c (CSC)

Are all applicable external inputs checked with respect to specified ranges prior to use?

Y/N/NA

AM.2.4.c (CSC)

Are conflicting requests and illegal combinations of all applicable inputs identified and checked?

Y/N/NA

AM.2.5.c (CSC)

Are all inputs checked and all errors (resulting from those inputs) reported before processing begins?

Y/N/NA

AM.2.6.c (CSC)

Is there a check to see if all data is available before processing begins?

Y/N/NA

AM.3.1.c (CSCI)

Are there provisions for detection of and recovery from all computational failures (i.e., using exception handlers or other means)?

Y/N/NA

AM.3.2.c (CSCI)

Are there provisions to range-test critical loop and index parameters (by explicit checks in the code or by feature of the Ada language) before use?

Y/N/NA

AM.3.3.c (CSCI)

Are there provisions to range-test all critical (i.e., supporting a mission critical capability) subscript values before use?

Y/N/NA

AM.3.4.c (CSCI)

Are there provisions to check all critical outputs before final outputting?

Y/N/NA

AM.4.1.c (CSCI)

Are there provisions for recovery (i.e., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, hardware failure, clock interrupt)?

Y/N/NA

AM.5.1.c (CSCI)

Are there provisions for recovery from all I/O device errors (i.e., I/O Exceptions - Device Error)?

Y/N/NA

AM.6.1.c (CSCI)

Are there provisions (i.e., exception handlers or other means) for recovery from all communications transmission errors?

Y/N/NA

AM.6.2.c (CSC)

Is error checking information (for example, check sums, parity bit) computed and transmitted with all messages?

Y/N/NA

AM.6.3.c (CSC)

Is error checking information computed and compared upon receipt of all messages?

Y/N/NA

AM.6.4.c (CSC)

Are transmission retries limited for all messages?

Y/N/NA

AM.7.1.c (CSC)

Are there provisions for recovery from all failures to communicate with other nodes/systems?

Y/N/NA

AM.7.2.c (CSC)

Are there provisions to periodically check adjacent nodes or interoperating systems for operational status?

Y/N/NA

AM.7.3.c (CSC)

Are there provisions for alternate message routing?

Y/N/NA

AP.1.1.c (CSCI)

Does the design limit specific references to the database management scheme?

Y/N/NA

AP.3.1.c (CSCI)

Are specific references to the computer architecture localized (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.c (CSCI)

Is the CSCI free from microcode instructions?

Y/N/NA

AP.5.1.c (CSCI)

Did the design incorporate algorithms which were used by other CSCI applications? (For example, table-driven algorithms.)

Y/N/NA

AP.5.2.c (CSC)

How many CSC capabilities are there?

__/NA

AP.5.4.c (CSCI)

Have all algorithms been verified with respect to their requirements?

Y/N/NA

AP.5.5.c (CSCI)

Is there test data reflecting verification available for all algorithms?

Y/N/NA

AP.5.6.c (CSCI)

How many CSCs are there?

__/NA

AT.1.3.c (CSC)

What is the total memory space allocated?

__/NA

AT.1.4.c (CSC)

What is the estimated memory space used (total less spare capacity)?

__/NA

AT.1.6.c (CSC)

What is the total auxiliary space allocated?

__/NA

AT.1.7.c (CSC)

What is the estimated auxiliary space used (total less spare capacity)?

__/NA

AT.2.4.c (CSC)

What is the total processing time allocated?

__/NA

AT.2.5.c (CSC)

What is the estimated processing time used (total less spare capacity)?

__/NA

AT.3.2.c (CSC)

What is the total I/O channel time allocated?

__/NA

AT.3.3.c (CSC)

What is the estimated I/O channel time used (total less spare capacity)?

__/NA

AT.3.5.c (CSC)

What is the total communication channel time allocated?

__/NA

AT.3.6.c (CSC)

What is the estimated communication channel time used (total less spare capacity)?

__/NA

AT.4.1.c (CSCI)

Are all processors, communication links, memory devices, and peripherals compatible (e.g., of a common vendor or model)?

Y/N/NA

AU.2.2.c (CSCI)

Does the executive software perform testing of its own operation, communication links, memory devices, and peripheral devices?

Y/N/NA

CL.1.3.c (CSC)

Does this CSC receive input from other systems?

Y/N/NA

CL.1.4.c (CSC)

Does this CSC transmit output to other systems

Y/N/NA

CL.1.6.c (CSCI)

Is there compliance with network communication protocol standards?

Y/N/NA

CL.1.7.c (CSCI)

Is all network processing controlled in accordance with the protocol standards?

Y/N/NA

CL.1.8.c (CSCI)

Are all user sessions controlled in accordance with the protocol standards?

Y/N/NA

CL.1.10.c (CSCI)

Is all communication routing performed in accordance with the protocol standards?

Y/N/NA

CL.1.11.c (CSCI)
Are messages (e.g., synchronization, message decoding) handled in a uniform manner?
Y/N/NA

CL.1.12.c (CSCI)
Is all message handling (e.g., synchronization, message decoding) performed in a uniform manner in accordance with the protocol standards?
Y/N/NA

CL.1.17.c (CSCI)
Is there a description of how the computer system will appear to its users and how the users will interact with the system?
Y/N/NA

CL.1.18.c (CSCI)
Is there a complete and definitive set of operating procedures for using this system?
Y/N/NA

CL.1.19.c (CSCI)
Is the CSCI free from constraints with respect to external communication?
Y/N/NA

CL.2.2.c (CSC)
Do all data representations and translations between representations of data comply with the established standard?
Y/N/NA

CL.2.3.c (CSC)
Does the CSC perform data translations between representations of data?
Y/N/NA

CL.2.5.c (CSCI)
How many different formats are used for input data from other systems (e.g., formats for data positioning, data packing, block transmission)?
__/NA

CL.2.7.c (CSCI)
How many different formats are used to output data to other systems (e.g., formats for data positioning, data packing, block transmission)?
__/NA

CL.2.8.c (CSCI)

How many different types of input records are received from other systems?

__/NA

CL.2.9.c (CSCI)

How many types of input records transmitted from other systems contain tags identifying the type of data they contain?

__/NA

CL.2.10.c (CSCI)

How many different types of output records are transmitted to other systems?

__/NA

CL.2.11.c (CSCI)

How many types of output records to other systems contain tags identifying the type of data they contain?

__/NA

CP.1.1.c (CSC)

Are the inputs, processing, and outputs of the CSC specified?

Y/N/NA

CP.1.2.c (CSC)

How many data items are identified?

__/NA

CP.1.3.c (CSC)

How many identified data items are defined (documented with regard to their source, meaning, and format)?

__/NA

CP.1.4.c (CSC)

How many identified data items are defined, computed, or obtained from an external source? (For example, referencing: global data with preassigned values, input parameters with preassigned values.)

__/NA

CP.1.5.c (CSC)

How many data items are referenced?

__/NA

CP.1.6.c (CSC)

How many defined data items are referenced?

__/NA

CP.1.7.c (CSCI)

Have all capabilities for this CSCI been allocated to CSCs?

Y/N/NA

CP.1.10.c (CSC)

Are all conditions and alternative processing options defined for each decision point?

Y/N/NA

CP.1.12.c (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.c (CSCI)

How many recorded problem reports have been closed to date?

__/NA

CS.1.2.c (CSCI)

Do the design representations comply with the established standard? (For example, HIPO Charts, Program Design Language, Data Flow Diagrams.)

Y/N/NA

CS.1.5.c (CSCI)

Does the design comply with the standards established for the calling sequence protocol between CSUs?

Y/N/NA

CS.1.8.c (CSCI)

Does the design comply with the established standard for the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.11.c (CSCI)

Does the design comply with the established standard for error handling?

Y/N/NA

CS.1.12.c (CSCI)

For how many CSCs do all references to that CSC use a single, unique name?

__/NA

CS.2.2.c (CSCI)

Does the design comply with the established standard for data representation?

Y/N/NA

CS.2.5.c (CSC)

Do the data names in this CSC comply with the established standard?

Y/N/NA

CS.2.8.c (CSC)

Do the definitions and uses of global variables comply with the established standard?

Y/N/NA

CS.2.11.c (CSCI)

Are there procedures to establish consistency and concurrency of multiple copies of the same information? (For example, copies at different nodes.)

Y/N/NA

CS.2.13.c (CSCI)

Are there procedures to verify consistency and concurrency of multiple copies of the same information?

Y/N/NA

CS.2.14.c (CSC)

Do all references to the same data use a single, unique name?

Y/N/NA

DI.1.1.c (CSCI)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all CSCs and their interfaces in the CSCI?

Y/N/NA

DI.1.2.c (CSC)

Is a graphic portrayal (figures, diagrams, tables) provided, which identifies the different types of CSC information and the information flow within the CSC?

Y/N/NA

DI.1.3.c (CSCI)

Is information organized and distributed within the CSCI? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.c (CSCI)

Are all files/libraries accessible from each node in accordance with requirements?

Y/N/NA

DI.1.5.c (CSCI)

Can alternate processing sources be selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.6.c (CSCI)

Are all mission-critical capabilities distributed over redundant elements/nodes?

Y/N/NA

DI.1.8.c (CSCI)

Does the design distribute control capabilities across different nodes/elements to ensure system operation under anomalous conditions?

Y/N/NA

DI.1.9.c (CSCI)

Does the design distribute CSCI capabilities across several physically separated components that make up the distributed architecture?

Y/N/NA

DI.1.10.c (CSCI)

Can each node communicate with all remaining nodes, in accordance with the specified requirements?

Y/N/NA

DO.1.1.c (CSCI)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.c (CSCI)

Is the documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.2.c (CSCI)

Does the design documentation clearly depict control and data flow (e.g., graphic portrayal with accompanying explanation, PDL)?

Y/N/NA

DO.2.3.c (CSCI)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.4.c (CSCI)

Does all the software documentation have separate volumes or separations within a single volume based on CSCI capabilities?

Y/N/NA

DO.2.5.c (CSCI)

Does the documentation completely characterize the operational capabilities of the software? (For example, identify all the performance parameters and limitations.)

Y/N/NA

DO.2.6.c (CSCI)

Does the documentation include functional interfaces, function processing, and function algorithms for all identified CSCI capabilities?

Y/N/NA

DO.2.7.c (CSCI)

Does the documentation contain descriptions of all algorithms used and limitations, including inputs, outputs, and required precision?

Y/N/NA

EC.1.1.c (CSC)

Have performance requirements and limitations for system communication efficiency been specified for each CSC capability?

Y/N/NA

EP.1.1.c (CSC)

Have performance requirements and limitations for processing efficiency been specified for each CSC capability? (For example, flow time for processes, execution time.)

Y/N/NA

EP.1.9.c (CSCI)

Is the CSCI free from use of overlays?

Y/N/NA

EP.2.1.c (CSCI)

Has the storage of all information (e.g., files, code, arrays, buffers) been organized for efficient processing (e.g., minimum search time)?

Y/N/NA

EP.2.2.c (CSCI)

Does the source code language(s) enable variable initialization (at compile time) when the variable is declared? (Guaranteed value of "Y" for Ada.)

Y/N/NA

EP.2.8.c (CSCI)

Does the method(s) for relating similar data items facilitate efficient processing? (For example, arrays, doubly-linked lists, directories.)

Y/N/NA

ES.1.2.c (CSCI)

Does the memory management of the CSCI incorporate virtual storage?

Y/N/NA

ES.1.3.c (CSCI)

Does the memory management of the CSCI incorporate dynamic reallocation of physical memory space during execution (dynamic memory management)?

Y/N/NA

ES.1.11.c (CSCI)

Is the CSCI free from redundant storage of files and libraries? (For example, duplicate copies of files are not stored at different nodes, multiple versions of the same file are not part of the wcrking CSCI.)

Y/N/NA

ES.1.12.c (CSCI)

Are variables in this CSCI read/written to/from a file to save memory space?

Y/N/NA

FS.2.1.c (CSCI)

Does the design implement the CSCI capabilities in such a way as to facilitate their use in other similar CSCI applications?

Y/N/NA

FS.2.4.c (CSC)

Are all inputs documented as to the specific use and limitations of the data?

Y/N/NA

FS.2.5.c (CSC)

Are all input/output formats specified and documented?

Y/N/NA

FS.2.6.c (CSC)

Are all outputs documented as to the specific use and interpretation of the data?

Y/N/NA

FS.2.7.c (CSC)

How many CSC capabilities were extracted from the requirements of other applications?

__/NA

FS.3.1.c (CSCI)

Can the user choose computation and output options? (For example, user selecting type of coordinate system, output media, and format.)

Y/N/NA

FS.3.2.c (CSCI)

Can the resources allocated to capabilities be modified? (For example, changing the amount of memory work space for a capability.)

Y/N/NA

GE.1.1.c (CSCI)

Are components incorporated or referenced from other systems or libraries?

Y/N/NA

GE.2.1.c (CSCI)

Is the CSCI organized such that CSUs perform single processing capabilities?

Y/N/NA

GE.2.3.c (CSCI)

Is the CSCI free from machine-dependent operations?

Y/N/NA

GE.2.4.c (CSCI)

Does the design provide for changing the volume of data handled by the system without effecting the executable code?

Y/N/NA

GE.2.5.c (CSCI)

Does the design provide for ranges of data input values being changed without effecting the executable code? (For example, no error tolerances or range tests.)

Y/N/NA

ID.1.5.c (CSC)

Is the software free from using any nonstandard constructs of the implementation language (e.g., pragmas, predefined library programs)?

Y/N/NA

ID.2.1.c (CSCI)

Are the same version and dialect of the implementation language(s) supported on other machines? (Guaranteed "Y" for Ada if validated Ada compiler is used.)

Y/N/NA

MO.1.2.c (CSC)

Is the CSC developed according to structured design techniques?

Y/N/NA

MO.1.3.c (CSC)

Does the CSC have a single processing objective (i.e., all the processes within the CSC are related to the same objective)?

Y/N/NA

MO.2.2.c (CSCI)

How many interfaces among CSCs in this CSCI?

__/NA

MO.2.3.c1 (CSCI)

How many CSC interfaces include content coupling?

__/NA

MO.2.3.c2 (CSCI)

How many CSC interfaces include common coupling?

__/NA

MO.2.3.c3 (CSCI)

How many CSC interfaces include external coupling?

__/NA

MO.2.3.c4 (CSCI)

How many CSC interfaces include control coupling?

__/NA

MO.2.3.c5 (CSCI)
How many CSC interfaces include stamp coupling?

__/NA

MO.2.3.c6 (CSCI)
How many CSC interfaces include data coupling?

__/NA

MO.2.5.c (CSC)
What is the cohesion value of the CSC? (Functional = 1.0, Informational = 0.7, Communicational = 0.5, Procedural = 0.3, Classical = 0.1, Logical = 0.1, Coincidental = 0.0)

__/NA

OP.1.1.c (CSCI)
Does the design implement the specified operating characteristics of the CSCI (i.e., the normal and alternate procedures and actions performed by the CSCI)?

Y/N/NA

OP.1.2.c (CSCI)
Are all the error conditions reported to the operator/user as specified in the requirements?

Y/N/NA

OP.1.3.c (CSCI)
Is the capability provided for the operator/user to respond to all reported errors, as specified in the requirement?

Y/N/NA

OP.1.4.c (CSCI)
Can the operator interrupt system operation, save and enter data, and continue processing?

Y/N/NA

OP.1.6.c (CSCI)
Are the procedures specified for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.c (CSCI)
Is a hard copy log provided of all operator interactions with the system/CSCI?

Y/N/NA

OP.1.10.c (CSCI)

Are all access violations reported to the operator in accordance with the requirements?

Y/N/NA

OP.1.11.c (CSCI)

Are appropriate responses performed/provided for all access violations in accordance with requirements?

Y/N/NA

OP.1.12.c (CSCI)

Can the operator/software obtain specific system (or network) resource status information, and reallocate resources?

Y/N/NA

OP.1.13.c (CSCI)

Can the operator/user select different nodes for different types of processing, or for retrieval of different information?

Y/N/NA

OP.1.14.c (CSCI)

Can the operator/user manipulate data regardless of the data's location in the system?

Y/N/NA

OP.1.15.c (CSCI)

Are system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.6.c (CSCI)

Can the user review and modify all input data prior to execution?

Y/N/NA

OP.2.7.c (CSCI)

Are all user-input data terminated by explicitly defined logical end of input?

Y/N/NA

OP.2.8.c (CSCI)

Can the user select among options for input media (e.g., terminal, tape drive, card reader)?

Y/N/NA

OP.3.1.c (CSCI)
Can the user control output (e.g., choose specific outputs, output media, output formats, amount of output)?

Y/N/NA

OP.3.2.c (CSCI)
Does the design provide all outputs to the user with unique descriptive labels for identifying the data?

Y/N/NA

OP.3.3.c (CSCI)
Does the design provide all output to the user in user-oriented measurement units (e.g., pages, columns, rows, seconds, minutes, hours)?

Y/N/NA

OP.3.4.c (CSCI)
How many different formats are output to the user? (For example, CRT display arrangements, printer outputs.)

__/NA

OP.3.5.c (CSCI)
Are all user outputs separated into logical groups to facilitate user examination?

Y/N/NA

OP.3.6.c (CSCI)
Do all error messages identify the nature of the error in accordance with a project-established taxonomy?

Y/N/NA

OP.3.7.c (CSCI)
Can the user select among options for output media?

Y/N/NA

OP.3.8.c (CSCI)
Is there a standard (common) command language for network information and data access?

Y/N/NA

OP.3.9.c (CSCI)
Do all outputs to the user have unique descriptive labels for identifying data?

Y/N/NA

OP.3.10.c (CSCI)

Are all outputs to the user provided with user-oriented measurement units?

Y/N/NA

RE.1.1.c (CSCI)

Do communication paths exist to all remaining nodes/communication links in the event of a failure of one node/link?

Y/N/NA

RE.1.2.c (CSCI)

Is the integrity of all data values maintained following the occurrence of anomalous conditions?

Y/N/NA

RE.1.3.c (CSCI)

Can all disconnected nodes rejoin the network after recovery, such that the processing capabilities of the system are not interrupted?

Y/N/NA

RE.1.4.c (CSCI)

Are all critical data replicated at two or more distinct nodes, in accordance with specified requirements?

Y/N/NA

SD.3.7.c (CSC)

Is the CSC structured in the standard established format?

Y/N/NA

SI.1.1.c (CSCI)

Does the design of the CSCI reflect a structured design approach? (For example, top-down design.)

Y/N/NA

SI.1.7.c (CSCI)

How many unique data items are in common blocks in this CSCI? (Guaranteed "NA" for Ada software.)

__/NA

SI.1.8.c (CSCI)

How many unique common blocks in this CSCI? (Guaranteed "NA" for Ada software.)

__/NA

SI.1.9.c (CSCI)

Are there requirements for a programming standard?

Y/N/NA

SI.1.10.c (CSCI)

Has a programming standard been established?

Y/N/NA

SI.1.11.c (CSC)

Does this CSC description identify all interfacing CSCs and interfacing hardware?

Y/N/NA

SI.2.1.c (CSCI)

Are there requirements to use a structured language or preprocessor for CSCI implementation?

Y/N/NA

SS.1.1.c (CSCI)

Are there controls on user input/output access, in accordance with the specified requirements? (For example, user access is limited by identification and password checking.)

Y/N/NA

SS.1.2.c (CSCI)

Are there controls on data access in the CSCI, in accordance with the specified requirements? (For example, authorization tables and privacy locks.)

Y/N/NA

SS.1.3.c (CSCI)

Are there controls the scope of task operations during execution, in accordance with the specified requirements? (For example, invoke other tasks, access system registers, or use privileged commands.)

Y/N/NA

SS.1.4.c (CSCI)

Are there controls on access to the network, in accordance with the specified requirements?

Y/N/NA

SS.2.1.c (CSCI)

Is all access to the system recorded and reported in accordance with the specified requirements? (For example, terminal and processor linkage, data file access, and jobs run information.)

Y/N/NA

SS.2.2.c (CSCI)

Are all access violations immediately indicated and identified in accordance with the specified requirements?

Y/N/NA

ST.3.1.c (CSCI)

Is I/O isolated from computations?

Y/N/NA

ST.3.2.c (CSCI)

Does each CSC perform unique operations (i.e., similar operations are not performed within different CSCs which could be restricted to a single CSC)?

Y/N/NA

ST.3.3.c (CSC)

How many non-related capabilities are performed in the CSC (i.e., capabilities which do not contribute to the same overall objective)?

__/NA

SY.1.1.c (CSCI)

Does the CSCI use the same I/O transmission rate as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.2.c (CSCI)

Does the CSCI use the same communication protocol as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.3.c (CSCI)

Is there a common interpretation of the content in all messages sent from and received by this system/CSCI and the interoperating system(s), in accordance with the specified requirements? (For example, all variables in the message have the same meaning.)

Y/N/NA

SY.1.4.c (CSCI)

Does the CSCI use the same structure and sequence for message contents as the interoperating system(s), in accordance with the specified requirements? (For example, all real variables are 16 bits in length, and real coordinates are ordered Xcoord, Ycoord, and Zcoord.)

Y/N/NA

SY.2.1.c (CSCI)

Does the CSCI use the same data format as the interoperating system(s), in accordance with the specified requirements? (For example, all characters are represented in ASCII format.)

Y/N/NA

SY.2.2.c (CSCI)

Does the CSCI use the same data base structure as the interoperating system(s), in accordance with the specified requirements? (For example, all systems use a relational data base containing similar information.)

Y/N/NA

SY.2.3.c (CSCI)

Does the CSCI provide the same data base access techniques as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.3.3.c (CSCI)

Does the CSCI use the same instruction set as the interoperating system(s)?

Y/N/NA

SY.4.1.c (CSCI)

Does the CSCI use the same source code language(s) as the interoperating system(s)?

Y/N/NA

SY.4.2.c (CSCI)

Does this CSCI use the same operating system as the interoperating system(s)?

Y/N/NA

SY.4.3.c (CSCI)

Does this CSCI use the same support software as the interoperating system(s)?

Y/N/NA

SY.5.1.c (CSCI)

Is documentation from the interoperating system(s) available that enables interoperability requirements to be established for this system? (For example, documentation is up-to-date, complete, and clearly organized.)

Y/N/NA

TC.1.1.c (CSCI)

Is there a table(s) tracing all CSC allocated requirements to the parent CSCI requirements specification?

Y/N/NA

TC.1.2.c (CSC)

Does the description of the CSC identify all the specified requirements (at the CSCI level) the CSC helps satisfy?

Y/N/NA

VS.1.4.c (CSC)

Is there a requirement to test all input parameters?

Y/N/NA

VS.2.2.c (CSCI)

Are interfaces among the CSCs to be tested?

Y/N/NA

VS.3.1.c (CSCI)

Are specified performance requirements be tested?

Y/N/NA

VS.3.2.c (CSCI)

Are all CSCs of the CSCI to be exercised during CSCI testing?

Y/N/NA

VS.3.3.c (CSCI)

Is there provision for a summary table listing all inputs and outputs for testing?

Y/N/NA

VS.4.1.c (CSCI)

Are CSCI capability requirements to be tested?

Y/N/NA

DATA COLLECTION FORM -- D-LEVEL

AC.1.3.d (CSU)

Do the numerical techniques used in the CSU implementing mission-critical capabilities provide enough precision to support accuracy objectives?

Y/N/NA

AC.1.4.d (CSU)

Are there quantitative accuracy requirements for all applicable inputs associated with each mission critical capability?

Y/N/NA

AC.1.5.d (CSU)

Are there quantitative accuracy requirements for all applicable outputs associated with each mission critical capability?

Y/N/NA

AC.1.6.d (CSU)

Are there quantitative accuracy requirements for all applicable constants associated with each mission critical capability?

Y/N/NA

AC.1.7.d (CSU)

Do the existing math library subprograms which are planned for use in the CSU provide enough precision to support accuracy objectives?

Y/N/NA

AM.1.5.d (CSU)

When an error condition is detected, is its resolution determined by the calling CSU?

Y/N/NA

AM.2.1.d (CSC)

Are error tolerances specified for all particular external input data (e.g., range of numerical values, legal combinations of alphanumeric values)?

Y/N/NA

AM.2.2.d (CSU)

Are values of all applicable inputs range-specified?

Y/N/NA

AM.2.3.d (CSU)

Are all applicable external inputs checked with respect to specified ranges prior to use?

Y/N/NA

AM.2.4.d (CSU)

Are all applicable external inputs checked with respect to illegal combinations and conflicting requests prior to use?

Y/N/NA

AM.2.5.d (CSU)

Are all detected errors with respect to applicable external inputs reported before processing begins?

Y/N/NA

AM.2.6.d (CSU)

Are all applicable external inputs checked for reasonableness before processing begins?

Y/N/NA

AM.3.1.d (CSU)

Is recovery provided for all computational failures within the CSU?

Y/N/NA

AM.3.2.d (CSU)

Are all critical (i.e., supporting a mission-critical capability) parameters checked by explicit checks in the code or by features of the Ada language for out-of-range values before use?

Y/N/NA

AM.3.3.d (CSU)

Are all critical (i.e., supporting a mission-critical capability) subscript values checked for out-of-range values before use?

Y/N/NA

AM.3.4.d (CSU)

Are all critical (i.e., supporting a mission-critical capability) output data checked for reasonable values before final outputting?

Y/N/NA

AM.4.1.d (CSC)

Is recovery made (e.g., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, hardware failure, clock interrupt)?

Y/N/NA

AM.5.1.d (CSC)

Are there provisions for recovery from all I/O device errors (i.e., I/O Exceptions - Device Error)?

Y/N/NA

AP.1.1.d (CSU)

Is the CSU free from references to the database management scheme?

Y/N/NA

AP.2.1.d (Procedure)

How many unique parameters are in the procedure?

__/NA

AP.2.2.d (Procedure)

How many global variables are referenced by the procedure? (a global variable is defined as a variable declared or visible in a procedure other than that containing the code referencing the variable).

__/NA

AP.3.1.d (CSC)

Are specific references to the computer architecture localized (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.d (Procedure)

Is the procedure free from microcode instruction statements?

Y/N/NA

AP.5.1.d (CSC)

Does the design incorporate algorithms which are not unique to this CSC's application? (For example, table-driven algorithms.)

Y/N/NA

AP.5.2.d (CSCI)

How many CSCs are there?

__/NA

AP.5.6.d (SYSTEM)

How many CSCIs are there?

__/NA

AP.5.7.d (CSC)

How many CSUs are there?

__/NA

AT.1.1.d (CSU)

Are all variable dimensions and dynamic array sizes defined parametrically for this CSU?

Y/N/NA

AT.1.3.d (CSC)

What is the total memory space allocated?

__/NA

AT.1.4.d (CSC)

What is the estimated memory space used (total less spare capacity)?

__/NA

AT.1.6.d (CSC)

What is the total auxiliary space allocated?

__/NA

AT.1.7.d (CSC)

What is the estimated auxiliary space used (total less spare capacity)?

__/NA

AT.2.1.d (CSU)

Are all accuracy, convergence, timing attributes, and timing limitations defined parametrically for this CSU?

Y/N/NA

AT.2.2.d (CSU)

Does this CSU use tables to determine processing decisions performed by the CSU?

Y/N/NA

AT.2.4.d (CSC)

What is the total processing time allocated?

__/NA

AT.2.5.d (CSC)

What is the estimated processing time used (total less spare capacity)?

__/NA

AT.3.2.d (CSC)
What is the total I/O channel time allocated?

__/NA

AT.3.3.d (CSC)
What is the estimated I/O channel time used (total less spare capacity)?

__/NA

AT.3.5.d (CSC)
What is the total communications time allocated (total less spare capacity)?

__/NA

AT.3.6.d (CSC)
What is the estimated communication channel time used (total less spare capacity)?

__/NA

AU.1.2.d (Procedure)
How many lines of source code are there, excluding comments and blank lines?

__/NA

AU.1.3.d (Procedure)
How many lines of source code are there to handle hardware/device interface protocol?

__/NA

AU.1.5.d (CSU)
Is the CSU required to perform hardware/device interface protocol?

Y/N/NA

AU.1.6.d (CSCI)
What is the estimated CPU processing time in seconds typically spent executing hardware/device interface protocol?

__/NA

AU.1.7.d (CSCI)
What is the estimated CPU processing time in seconds typically spent executing the entire CSCI?

__/NA

CL.1.3.d (CSU)
Does the CSU receive input from other systems?

Y/N/NA

CL.1.4.d (CSU)

Does this CSU transmit output to other systems?

Y/N/NA

CL.2.2.d (CSU)

Do all data representations and translations between representations of data comply with the established standard?

Y/N/NA

CL.2.3.d (CSU)

Does the CSU perform data translation between representations of data?

Y/N/NA

CP.1.1.d (CSU)

Are the inputs, processing, and outputs of the CSU specified?

Y/N/NA

CP.1.2.d (Procedure)

How many data items are identified?

__/NA

CP.1.3.d (Procedure)

How many identified data items are defined (documented with regard to their source, meaning, and format)?

__/NA

CP.1.4.d (Procedure)

How many identified data items are defined, computed, or obtained from an external source? (For example, referencing: global data with preassigned values, input parameters with preassigned values.)

__/NA

CP.1.5.d (Procedure)

How many data items are referenced?

__/NA

CP.1.6.d (Procedure)

How many defined data items are referenced?

__/NA

CP.1.7.d (CSC)

Have all capabilities for this CSC been allocated to CSUs?

Y/N/NA

CP.1.10.d (CSU)

Are all conditions and alternative processing options defined for each decision point in the CSU?

Y/N/NA

CP.1.11.d (Procedure)

Are all parameters in the argument list used in the procedure?

Y/N/NA

CP.1.12.d (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.d (CSCI)

How many recorded problem reports have been corrected to date?

__/NA

CS.1.2.d (CSCI)

Does the design representations comply with the established standard? (For example, HIPO Charts, Program Design Language, Data Flow Diagrams.)

Y/N/NA

CS.1.5.d (CSU)

Does the CSU comply with the standards established for the calling sequence protocol between CSUs?

Y/N/NA

CS.1.8.d (CSU)

Does the CSU comply with the established standard for the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.11.d (CSU)

Does the CSU comply with the established standard for error handling?

Y/N/NA

CS.1.12.d (CSC)

For how many CSUs do all references to that CSU use a single, unique name?

__/NA

CS.2.2.d (CSU)

Does the data representation comply with the established standard?

Y/N/NA

CS.2.5.d (CSU)

Do the data names in this CSU comply with the established standard?

Y/N/NA

CS.2.8.d (CSU)

Do the definitions and uses of global variables comply with the established standard?

Y/N/NA

CS.2.14.d (CSU)

Do all references to the same data use single unique names?

Y/N/NA

DI.1.1.d (CSCI)

Are graphic portrayals (figures, diagrams, tables) provided which identify the decomposition of all CSCs into CSUs?

Y/N/NA

DI.1.2.d (CSCI)

Is a graphic portrayal (figures, diagrams, tables) provided which identifies all the different types of CSCI information and the information flow within the CSCI?

Y/N/NA

DI.1.3.d (CSCI)

Is information organized and distributed within the CSCI? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.d (CSCI)

Are all files/libraries accessible from each node in accordance with requirements?

Y/N/NA

DI.1.5.d (CSCI)

Can alternate processing sources be selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.6.d (CSCI)
Are all mission-critical capabilities distributed over redundant elements/nodes?
Y/N/NA

DI.1.8.d (CSCI)
Are all control capabilities distributed across different nodes/elements to ensure system operation under anomalous conditions?
Y/N/NA

DI.1.9.d (CSCI)
Are CSCI capabilities implemented across several physically separated components that make up the distributed architecture?
Y/N/NA

DI.1.10.d (CSCI)
Can each node communicate with all remaining nodes, in the event a node becomes unavailable, in accordance with the specified requirements?
Y/N/NA

DO.1.1.d (CSCI)
Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?
Y/N/NA

DO.2.1.d (CSCI)
Is the documentation structured such that capabilities are separately specified?
Y/N/NA

DO.2.2.d (CSCI)
Does the design documentation clearly depict control and data flow (e.g., graphic portrayal with accompanying explanation, PDL)?
Y/N/NA

DO.2.3.d (CSCI)
Does each document contain a scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs, index.)
Y/N/NA

DO.2.4.d (CSCI)

Do all the software specifications and design and test documentation have separate volumes or separations within a single volume based on system capabilities, CSCI capabilities, software capabilities, or software elements?

Y/N/NA

DO.2.5.d (CSCI)

Does the documentation completely characterize the operational capabilities of the software? (For example, identify all the performance parameters and limitations.)

Y/N/NA

DO.2.6.d (CSC)

Does the documentation include functional interfaces, function processing, and function algorithms for all identified CSC capabilities?

Y/N/NA

DO.2.7.d (CSCI)

Does the documentation contain descriptions of all algorithms used and limitations, including inputs, outputs, and required precision?

Y/N/NA

EP.1.1.d (CSU)

Is the unit required to be optimized for processing efficiency?

Y/N/NA

EP.1.5.d (Procedure)

How many loops are in the procedure (WHILE loops, REPEAT UNTIL loops, and iteration loops)?

__/NA

EP.1.6.d (Procedure)

How many loops contain non-loop dependent statements? (For example, initializing or calculating a variable which is not related to any values which change within the loop.)

__/NA

EP.1.7.d (Procedure)

How many instances are there of 2 or more operations in an expression (compound expression)?

__/NA

EP.1.8.d (Procedure)

How many compound expressions are recalculated needlessly (all variables in the expression have not been reassigned values)?

__/NA

EP.1.10.d (Procedure)

How many instances of bit/byte packing/unpacking are performed?

__/NA

EP.1.11.d (Procedure)

How many instances of bit/byte packing/unpacking are performed needlessly in a loop (i.e., could be performed outside the loop)?

__/NA

EP.2.1.d (CSCI)

Has the storage of all information (e.g., files, code, arrays, buffers) been organized for efficient processing (e.g., minimum search time)?

Y/N/NA

EP.2.3.d (Procedure)

How many arithmetic expressions in the procedure?

__/NA

EP.2.4.d (Procedure)

How many arithmetic expressions with different sized components in the same expression? (For example, byte/word/double word.)

__/NA

EP.2.5.d (Procedure)

How many arithmetic expressions with mixed data types in the same expression? (For example, integer/real/boolean/literal.)

__/NA

EP.2.7.d (Procedure)

How many data items are modified in the procedure?

__/NA

EP.2.8.d (CSCI)

Does the method(s) for relating similar data items facilitate efficient processing? (For example, arrays, doubly-linked lists, directories.)

Y/N/NA

ES.1.5.d (CSCI)

How many global variables are in the CSCI? (a global variable is defined as a variable declared or visible in a CSCI other than that containing the code referencing the variable).

__/NA

ES.1.6.d (CSCI)

How many global variables are referenced by more than one name?

__/NA

ES.1.7.d (CSCI)

Does the separation of the CSCI into segments (load modules) efficiently utilize the segmented memory space available? (For example, minimizing the largest segment length to minimize the memory segment size required for module execution.)

Y/N/NA

ES.1.8.d (Procedure)

Are there any data packing operations in the procedure?

Y/N/NA

ES.1.11.d (CSCI)

Is the CSCI free from redundant storage of files and libraries? (For example, duplicate copies of files are not stored at different nodes, multiple versions of the same file are not part of the working CSCI.)

Y/N/NA

FS.1.1.d (CSU)

Does this CSU perform a single capability?

Y/N/NA

FS.2.1.d (CSC)

Does the design implement the CSC capabilities in such a way as to facilitate their use in other similar CSC applications?

Y/N/NA

GE.1.1.d (CSCI)

How many CSUs are called by more than one other CSU?

__/NA

GE.2.1.d (CSU)

In the CSU, are the following processing categories mixed: External Input, External Output, or Algorithmic Processing?

Y/N/NA

GE.2.3.d (CSU)

Is this CSU free from machine-dependent operations?

Y/N/NA

GE.2.4.d (CSU)

Is this CSU free from strict limitations on the volume of data items it processes? (For example, data volume limits are parameterized.)

Y/N/NA

GE.2.5.d (CSU)

Is this unit free from strict limitations on the values of input data? (For example, no error tolerances are specified, no range-tests or reasonableness checks are performed.)

Y/N/NA

ID.1.3.d (Procedure)

How many references are there to system library routines, utilities, or other system provided facilities?

__/NA

ID.1.5.d (Procedure)

Is the procedure design free from using any nonstandard constructs of the implementation language(s)?

Y/N/NA

ID.2.3.d (Procedure)

Does the procedure perform external input or output?

Y/N/NA

ID.2.4.d (Procedure)

Does the procedure contain operations dependent on word or character size?

Y/N/NA

ID.2.5.d (Procedure)

Does the procedure contain data element representations that are machine dependent?

Y/N/NA

MO.1.2.d (CSU)

Is the CSU developed according to structured design techniques?

Y/N/NA

MO.1.3.d (CSU)
Does this CSU have a single processing objective (i.e., all processing within the unit is related to the same objective)?

Y/N/NA

MO.1.4.d (CSU)
Are the estimated lines of source code for this CSU (excluding comment lines and blank lines) 100 lines or less?

Y/N/NA

MO.1.5.d (Procedure)
How many unique parameters are in the procedure?

__/NA

MO.1.6.d (Procedure)
How many calling sequence parameters are control variables (select an operating mode or submode in the unit, direct the sequential flow, or otherwise directly influence the function of the procedure)?

__/NA

MO.1.7.d (Procedure)
Is all input data passed into the procedure through calling sequence parameters (i.e., no data is input through global areas or input statements)?

Y/N/NA

MO.1.8.d (Procedure)
Is output data passed back to the calling unit? (For example, through calling sequence parameters or global areas.)

Y/N/NA

MO.1.9.d (Procedure)
Is control always returned to the calling procedure when execution is completed?

Y/N/NA

MO.1.10.d (CSU)
Is temporary storage (work space reserved for intermediate or partial results) used only by this unit during execution (i.e. not shared with other CSUs)?

Y/N/NA

MO.2.2.d (CSCI)
How many interfaces among CSUs in this CSCI?

__/NA

MO.2.3.d1 (CSCI)
How many CSU interfaces include content coupling?

__/NA

MO.2.3.d2 (CSCI)
How many CSU interfaces include common coupling?

__/NA

MO.2.3.d3 (CSCI)
How many CSU interfaces include external coupling?

__/NA

MO.2.3.d4 (CSCI)
How many CSU interfaces include control coupling?

__/NA

MO.2.3.d5 (CSCI)
How many CSU interfaces include stamp coupling?

__/NA

MO.2.3.d6 (CSCI)
How many CSU interfaces include data coupling?

__/NA

MO.2.5.d (CSU)
What is the cohesion value of the CSU? (Functional = 1.0, Informational = 0.7, Communicational = 0.5, Procedural = 0.3, Classical = 0.1, Logical = 0.1, Coincidental = 0.0)

__/NA

OP.1.8.d (CSCI)
How many different format types are used for operator messages and response?

__/NA

OP.1.9.d (CSCI)
How many total operator messages and responses are provided?

__/NA

OP.1.17.d (CSCI)
Are all specified error conditions reported to the operator/user such that the nature of the error and any response required by the operator/user are clearly identified and described in the error message?

Y/N/NA

OP.1.18.d (CSCI)

Is the capability provided for operator/user response to all reported errors as specified in the requirements?

Y/N/NA

SI.1.1.d (CSCI)

Does the design of the CSCI reflect a structured design approach? (For example, top-down design.)

Y/N/NA

SI.1.2.d (Procedure)

Is the procedure independent of the source of input and destination of output?

Y/N/NA

SI.1.3.d (CSU)

Is the CSU independent of knowledge of prior processing?

Y/N/NA

SI.1.4.d (CSU)

Does the CSU description/prologue include input, output, processing and limitations?

Y/N/NA

SI.1.5.d (Procedure)

How many entrances into the procedure?

__/NA

SI.1.6.d (Procedure)

How many exits from the procedure?

__/NA

SI.1.7.d1 (Procedure)

How many unique data items are in common blocks in this procedure?

__/NA

SI.1.7.d2 (CSCI)

How many unique data items are in common blocks in this CSCI?

__/NA

SI.1.8.d (CSCI)

How many unique common blocks in this CSCI?

__/NA

SI.1.11.d (CSU)

Does this CSU description identify all interfacing units and interfacing hardware?

Y/N/NA

SI.2.1.d (Procedure)

Is the procedure implemented in a structured language or using a preprocessor?

Y/N/NA

SI.3.1.d (Procedure)

How many conditional branch statements are there in the procedure? (For example, IF, WHILE, REPEAT, DO/FOR loop, CASE.)

__/NA

SI.3.2.d (Procedure)

How many unconditional branch statements are there in the procedure? (For example, GOTO, CALL, RETURN.)

__/NA

SI.4.1.d (Procedure)

Is the procedure's flow of control from top to bottom (i.e., control does not erratically jump)?

Y/N/NA

SI.4.3.d (Procedure)

How many negative boolean and compound boolean expressions are used in the procedure?

__/NA

SI.4.5.d (Procedure)

How many loops have unnatural exits? (For example, jumps out of loop, return statements.)

__/NA

SI.4.6.d (Procedure)

How many iteration loops are used in the procedure (DO/FOR loops)?

__/NA

SI.4.7.d (Procedure)

In how many iteration loops are indices modified to alter the fundamental processing of the loop?

__/NA

SI.4.8.d (CSU)

Is the CSU free from all self modification of code? (For example, the unit doesn't alter instructions, overlays of code, etc.)

Y/N/NA

SI.4.10.d (Procedure)

What is the maximum nesting level in the procedure?

__/NA

SI.4.11.d (Procedure)

How many total branches (conditional and unconditional) are used in the procedure?

__/NA

SI.4.12.d (Procedure)

How many data declaration statements are there in the procedure?

__/NA

SI.4.13.d (Procedure)

How many data manipulation statements are there in the procedure?

__/NA

SI.4.14.d (Procedure)

How many total data items (local and global) are used in the procedure?

__/NA

SI.4.15.d (Procedure)

How many data items are referenced locally in the procedure? (For example, variables declared locally and value parameters.)

__/NA

SI.4.16.d (CSU)

Does each data item in the CSU have a single use? (For example, each array serves only one purpose.)

Y/N/NA

SI.4.18.d (CSCI)

Is repeated and redundant code avoided by utilizing procedures, functions, generics, tasking, and packages?

Y/N/NA

SI.5.1.d (Procedure)

How many data items are used as input to the procedure?

__/NA

SI.5.2.d (Procedure)

How many data items are used for output by the procedure?

__/NA

SI.5.3.d (Procedure)

How many parameters in the procedure's calling sequence return output values?

__/NA

SI.5.4.d (CSU)

Does the unit perform a single, non-divisible capability?

Y/N/NA

SI.6.1.d (Procedure)

How many unique operators are in the procedure?

__/NA

SI.6.2.d (Procedure)

How many unique operands are in the procedure?

__/NA

SI.6.3.d (Procedure)

How many total operands are in the procedure?

__/NA

ST.1.1.d (Procedure)

How many data items are in this procedure's interface (i.e. data items used to input or output data to the procedure)?

__/NA

ST.1.2.d (Procedure)

How many global data items in this procedure's interface are not adequately commented (i.e., lack comments regarding the purpose, type, limitations of the data)?

__/NA

ST.1.3.d (Procedure)

How many interface data items are in the unit with negative qualification logic? (For example, boolean values that return "true" upon failure rather than success.)

__/NA

ST.1.4.d (CSU)

Is the CSU interface established solely by arguments in the calling sequence parameter list (i.e., is the CSU free from references to global data)?

Y/N/NA

ST.1.5.d (CSU)

Does the CSU modify the internal code or data of other CSUs?

Y/N/NA

ST.2.1.d (Procedure)

How many unique execution paths are in the procedure, including those causes by a "Raise" statement?

__/NA

ST.2.2.d (Procedure)

How many conditional branch statements are in the procedure?? (For example, IF, WHILE, REPEAT, CASE.)

__/NA

ST.2.3.d (Procedure)

How many other procedures are called by this procedure? (For example, calls to other functions, subroutines, procedures.)

__/NA

ST.2.4.d (Procedure)

How many iteration loops are there in the procedure? (For example, DO/FOR loops.)

__/NA

ST.2.5.d (CSU)

Are there comments regarding the CSUs called by this CSU and the units which call this CSU?

Y/N/NA

ST.3.2.d (CSCI)

Does the design isolate I/O capabilities from computational capabilities?

Y/N/NA

ST.3.4.d (CSU)

Is temporary storage (work space reserved for intermediate or partial results) used only by this CSU during execution (i.e., not shared with other CSUs)?

Y/N/NA

ST.3.5.d (CSU)

Does the CSU mix the management of primary and secondary storage resources with the management of data within the storage areas? (For example, an executive CSU that allocates storage for process, and controls what data can be accessed during process execution?)

Y/N/NA

ST.4.4.d (CSCI)

How many global data items are modified (set or changed) by one CSU and referenced by other CSUs? (a global variable is defined as a variable declared or visible in a CSU other than that containing the code referencing the variable).

__/NA

ST.4.5.d (Procedure)

Does this procedure have a single entrance (all procedures calling this procedure must enter at the same location)?

Y/N/NA

ST.4.6.d (CSU)

Does this CSU's communication with all interfacing CSUs pass only data parameters (i.e., does not pass any control elements)?

Y/N/NA

ST.5.1.d (CSU)

Is the CSU free from unnecessarily recomputing the same value?

Y/N/NA

ST.5.2.d (CSU)

Is the CSU free from statements which are never executed?

Y/N/NA

ST.5.3.d (CSU)

Is the meaning of each data item consistent throughout the CSU (i.e., the use associated with each data item does not change)?

Y/N/NA

ST.5.4.d (CSU)

Is the CSU free from unnecessary intermediate data items?

Y/N/NA

TC.1.1.d (CSC)

Is the decomposition of the CSCs into CSUs graphically depicted?

Y/N/NA

TC.1.2.d (CSU)

Does the description of the CSU identify all the specified requirements (at the CSC or CSCI level) that the CSU he!ps to satisfy?

Y/N/NA

TN.1.1.d (CSCI)

Have lesson plans and training materials been provided for operators, end users, and maintainers of the CSCI?

Y/N/NA

TN.1.2.d (CSCI)

Have realistic simulation exercises for the CSCI been provided?

Y/N/NA

TN.1.3.d (CSCI)

Are "HELP" information and diagnostic information provided for the operator, end user, and maintainer of the CSCI? (For example, an on-line list of legal commands or a list of the sequential steps in a process are provided.)

Y/N/NA

TN.1.4.d (CSCI)

Can CSCI users select the level of aid and guidance, according to their degree of expertise?

Y/N/NA

VS.1.1.d (Procedure)

How many execution paths are there in the procedure?

__/NA

VS.1.2.d (Procedure)

How many execution paths are to be tested?

__/NA

VS.1.3.d (Procedure)

How many total input parameters are there in the procedure?

__/NA

VS.1.4.d (Procedure)

How many input parameters are to be tested?

__/NA

VS.2.1.d (CSCI)

How many total interfaces are there between CSUs in the CSCI (e.g., one unit uses an item from another unit)?

__/NA

VS.2.2.d (CSCI)

How many CSU interfaces are to be tested?

__/NA

VS.3.1.d (CSCI)

Are all specified performance requirements of the CSCI to be tested?

Y/N/NA

VS.3.2.d (CSU)

Is the CSU to be exercised during CSCI testing?

Y/N/NA

VS.3.3.d (CSC)

Is there provision for a summary table listing all inputs and outputs for testing?

Y/N/NA

VS.4.1.d (CSC)

Are CSC capability requirements to be tested?

Y/N/NA

DATA COLLECTION FORM -- E-LEVEL

AC.1.5.e (CSU)

During execution, are all outputs within the specified accuracy tolerances?

Y/N/NA

AM.1.5.e (CSU)

When an error condition is detected, is its resolution determined by the calling CSU?

Y/N/NA

AM.2.1.e (CSU)

Are error tolerances specified for all particular external input data (e.g., range of numerical values, legal combinations of alphanumeric values)?

Y/N/NA

AM.2.3.e (CSU)

Are all applicable external inputs checked with respect to specified ranges before use?

Y/N/NA

AM.2.4.e (CSU)

Are all applicable external inputs checked with respect to illegal combinations and conflicting requests prior to use?

Y/N/NA

AM.2.5.e (CSU)

Are all detected errors with respect to applicable external inputs reported before processing begins?

Y/N/NA

AM.2.6.e (CSU)

Are all applicable external inputs checked for reasonableness before processing begins?

Y/N/NA

AM.3.1.e (CSU)

Is recovery provided for all computational failures within the CSU?

Y/N/NA

AM.3.2.e (CSU)

Are all critical (i.e., supporting a mission-critical capability) loop and index parameters checked by explicit checks in the code or by features of the Ada language for out-of-range values before use?

Y/N/NA

AM.7.2.e (CSC)

Are adjacent nodes or interoperating systems periodically checked for operational status?

Y/N/NA

AM.7.3.e (CSC)

Are there provisions for alternate message routing?

Y/N/NA

AP.1.1.e (CSU)

Is the CSU free from references to the database management scheme (e.g., all data calls for data base information are processed through an executive)?

Y/N/NA

AP.2.1.e (Procedure)

How many unique parameters are in the procedure?

__/NA

AP.2.2.e (Procedure)

How many global variables are referenced by the procedure? (a global variable is defined as a variable declared or visible in a procedure other than that containing the code referencing the variable).

__/NA

AP.2.3.e (CSU)

Do the comments for global data in the CSU explain where the data is derived, the data's composition, and how the data is used?

Y/N/NA

AP.2.4.e (CSU)

Does the unit contain comments for all parameter I/O and local variables describing each data item's composition and use?

Y/N/NA

AP.3.1.e (Procedure)

Is the unit free from computer architecture references?

Y/N/NA

AP.3.3.e (Procedure)

How many lines of code are there in the procedure, excluding comments and blank lines?

__/NA

AP.3.4.e (Procedure)
How many non-HOL lines of code are there in the procedure, excluding comment and blank lines? (For example, Assembly Language, Intermediate Code.)
__/NA

AP.4.1.e (Procedure)
Is the procedure free from microcode instruction statements?
Y/N/NA

AP.5.2.e (CSCI)
How many CSCs are there?
__/NA

AP.5.6.e (SYSTEM)
How many CSCIs are there?
__/NA

AP.5.7.e (CSC)
How manv CSUs are there?
__/NA

AT.1.1.e (CSU)
Are all array bounds, upper and lower (e.g., index constraints) defined parametrically?
Y/N/NA

AT.2.1.e (CSU)
Are all accuracy, convergence, timing attributes, and timing limitations defined parametrically for this CSU?
Y/N/NA

AT.2.2.e (CSU)
Are tables used in a manner which would ease the task of changing or expanding capabilities?
Y/N/NA

AU.1.3.e (Procedure)
How many lines of source code are there to handle hardware/device interface protocol?
__/NA

AU.1.5.e (CSU)
Is the CSU required to perform hardware/device interface protocol?
Y/N/NA

CL.1.3.e (CSU)
Does the CSU receive input from other systems?

Y/N/NA

CL.1.4.e (CSU)
Does the CSU transmit data to other systems?

Y/N/NA

CL.2.2.e (CSU)
Do all data representations and translations between representations of data comply with the established standard?

Y/N/NA

CL.2.3.e (CSU)
Does the unit perform data translation between representations of data?

Y/N/NA

CP.1.1.e (CSU)
Are the inputs, processing, and outputs of the CSU specified?

Y/N/NA

CP.1.2.e (Procedure)
How many data items are identified?

__/NA

CP.1.3.e (Procedure)
How many identified data items are defined (documented with regard to their source, meaning and format)?

__/NA

CP.1.4.e (Procedure)
How many identified data items are defined, computed, or obtained from an external source? (For example, referencing: global data with preassigned values, input parameters with preassigned values.)

__/NA

CP.1.5.e (Procedure)
How many data items are referenced?

__/NA

CP.1.6.e (Procedure)
How many defined data items are referenced?

__/NA

CP.1.10.e (CSU)

Are all conditions and alternative processing options defined for each decision point in the CSU?

Y/N/NA

CP.1.11.e (Procedure)

Are all parameters in the argument list used in the procedure?

Y/N/NA

CP.1.12.e (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.e (CSCI)

How many recorded problem reports have been corrected to date?

__/NA

CS.1.4.e (CSU)

Does the calling sequence comply with the established standard?

Y/N/NA

CS.1.5.e (CSU)

Does the CSU comply with the standards established for the calling sequence protocol between CSUs?

Y/N/NA

CS.1.8.e (CSU)

Does the CSU comply with the established standard for the external I/O protocol and format for all CSUs?

Y/N/NA

CS.1.11.e (CSU)

Does the handling of errors comply with the established standard?

Y/N/NA

CS.1.12.e (CSC)

For how many CSUs do all references to that unit use a single, unique name?

__/NA

CS.2.5.e (CSU)

Do the data names in this CSU comply with the established standard?

Y/N/NA

CS.2.8.e (CSU)

Do the definitions and uses of global variables comply with the established standard?

Y/N/NA

CS.2.14.e (CSU)

Do all references to the same data use single unique names?

Y/N/NA

DI.1.5.e (CSCI)

Can alternate processing sources be selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DO.1.1.e (CSCI)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.e (CSCI)

Is the documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.2.e (CSCI)

Does the program documentation clearly depict control and data flow (e.g., graphic portrayal with accompanying explanation, PDL)?

Y/N/NA

DO.2.3.e (CSCI)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.4.e (CSCI)

Does all the software documentation have separate volumes or separations within a single volume based on CSCI capabilities?

Y/N/NA

DO.2.6.e (CSU)

Does the documentation include functional interfaces, function processing, and function algorithms for all identified CSU capabilities?

Y/N/NA

EP.1.1.e (CSU)
Is the unit optimized for processing efficiency?

Y/N/NA

EP.1.3.e (CSU)
Is the CSU optimized for processing efficiency (i.e., compiled with an optimizing compiler or coded in assembly language)?

Y/N/NA

EP.1.5.e (Procedure)
How many loops are in the procedure (WHILE loops, REPEAT UNTIL loops, and iteration loops)?

__/NA

EP.1.6.e (Procedure)
How many loops contain non-loop dependent statements? (For example, initializing or calculating a variable which is not related to any values which change within the loop.)

__/NA

EP.1.7.e (Procedure)
How many instances are there of 2 or more operations in an expression (compound expression)?

__/NA

EP.1.8.e (Procedure)
How many compound expressions are recalculated needlessly (no variables in the expression have been reassigned values.)?

__/NA

EP.1.9.e (CSCI)
How many different overlays are used in the CSCI?

__/NA

EP.1.10.e (Procedure)
How many instances of bit/byte packing/unpacking are performed (i.e., use of the pragma PACK in Ada.)?

__/NA

EP.1.11.e (Procedure)
How many instances of bit/byte packing/unpacking are performed needlessly in a loop (i.e., could be performed outside the loop)? (Guaranteed score of 0 of pragma PACK is used in Ada.)

__/NA

EP.2.3.e (Procedure)

How many arithmetic expressions in the procedure?

__/NA

EP.2.4.e (Procedure)

How many arithmetic expressions with different sized components in the same expression? (For example, byte/word/double word.)

__/NA

EP.2.5.e (Procedure)

How many arithmetic expressions with mixed data types in the same expression? (For example, integer/real/boolean/literal.)

__/NA

EP.2.6.e (Procedure)

How many data items are in the procedure (arrays, constants, variables, etc.)?

__/NA

EP.2.7.e (Procedure)

How many data items are modified in the procedure?

__/NA

ES.1.5.e (CSCI)

How many global variables are in the CSCI? (a global variable is defined as a variable declared or visible in a CSCI other than that containing the code referencing the variable).

__/NA

ES.1.6.e (CSCI)

How many global variables are referenced by more than one name?

__/NA

ES.1.7.e (CSCI)

Does the separation of the CSCI into segments (load modules) efficiently utilize the segmented memory space available? (For example, minimizing the largest segment length to minimize the memory segment size required for module execution.)

Y/N/NA

ES.1.8.e (Procedure)

Are there any data packing operations in the procedure?

Y/N/NA

ES.1.9.e (CSU)

Is the CSU optimized for storage efficiency (i.e., compiled with an optimizing compiler or coded in assembly language)?

Y/N/NA

ES.1.11.e (CSCI)

Is the CSCI free from redundant storage of files and libraries? (For example, duplicate copies of files are not stored at different nodes, multiple versions of the same file are not part of the working CSCI.)

Y/N/NA

FS.1.1.e (CSU)

Does this CSU perform a single capability?

Y/N/NA

FS.1.2.e (CSU)

Is a description of the capability(s) performed provided in the CSU's comments?

Y/N/NA

GE.1.1.e (CSCI)

How many CSUs are called by more than one other CSU?

__/NA

GE.2.1.e (CSU)

Are the following processing categories mixed in this CSU: External Input, External Output, or Algorithmic Processing?

Y/N/NA

GE.2.3.e (CSU)

Is this CSU free from machine-dependent operations (e.g., no representation clauses, compiler predefined pragmas, or use of system-defined types)?

Y/N/NA

GE.2.4.e (CSU)

Can the volume of data processed by the unit be changed without effecting the executable code? (For example, data volume limits are parameterized.)

Y/N/NA

GE.2.5.e (CSU)

Can the range of data input be changed without effecting the executable code? (For example, no error tolerances are specified, no range-tests or reasonableness checks are performed.)

Y/N/NA

ID.1.1.e (Procedure)
Is a standard subset of the implementation language used?

Y/N/NA

ID.1.3.e (Procedure)
How many references are there to system library subprograms, utilities, or other system provided facilities?

__/NA

ID.1.5.e (Procedure)
Is the unit free from nonstandard constructs of the implementation language(s)?

Y/N/NA

ID.2.3.e (Procedure)
Does the procedure perform external input or output?

Y/N/NA

ID.2.4.e (CSU)
Does the CSU contain operations dependent on word or character size?

Y/N/NA

ID.2.5.e (Procedure)
Does the procedure contain data elements representations that are machine dependent?

Y/N/NA

MO.1.2.e (CSU)
Is the CSU coded and tested according to structured techniques? (For example, top-down implementation and testing.)

Y/N/NA

MO.1.3.e (CSU)
Does this CSU have a single processing objective (i.e., all processing within the CSU is related to the same objective)?

Y/N/NA

MO.1.4.e (Procedure)
Are the lines of source code for this procedure (excluding comment lines and blank lines) 100 lines or less?

Y/N/NA

MO.1.5.e (Procedure)

How many unique parameters are in the procedure?

__/NA

MO.1.6.e (Procedure)

How many calling sequence parameters are control variables (select an operating mode or submode in the procedure, direct the sequential flow, or otherwise directly influence the capability of the procedure)?

__/NA

MO.1.7.e (Procedure)

Is all input data passed into the procedure through calling sequence parameters (i.e., no data is input through global areas or input statements)?

Y/N/NA

MO.1.8.e (Procedure)

Is output data passed back to the calling procedure? (For example, through calling sequence parameters or global areas.)

Y/N/NA

MO.1.9.e (Procedure)

Is control always returned to the calling procedure when execution is completed?

Y/N/NA

MO.1.10.e (CSU)

Is temporary storage (work space reserved for intermediate or partial results) used only by this CSU during execution (i.e., not shared with other CSUs)?

Y/N/NA

MO.2.2.e (CSCI)

How many interfaces among CSUs in this CSCI?

__/NA

MO.2.3.e1 (CSCI)

How many CSU interfaces include content coupling?

__/NA

MO.2.3.e2 (CSCI)

How many CSU interfaces include common coupling?

__/NA

MO.2.3.e3 (CSCI)

How many CSU interfaces include external coupling?

__/NA

MO.2.3.e4 (CSCI)

How many CSU interfaces include control coupling?

__/NA

MO.2.3.e5 (CSCI)

How many CSU interfaces include stamp coupling?

__/NA

MO.2.3.e6 (CSCI)

How many CSU interfaces include data coupling?

__/NA

MO.2.5.e (CSU)

What is the cohesion value of this CSU?

__/NA

OP.1.8.e (CSCI)

How many different format types are used for operator messages and responses?

__/NA

OP.1.9.e (CSCI)

How many total operator messages and responses are provided?

__/NA

SD.1.2.e (Procedure)

How many comment lines are there in the procedure?

__/NA

SD.1.3.e (Procedure)

How many lines of source code with embedded comments?

__/NA

SD.2.1.e (CSU)

Does the CSU's prologue contain all the information in accordance with the established standard?

Y/N/NA

SD.2.2.e (CSU)

Is the identification and placement of all comments in the CSU in accordance with the established standard?

Y/N/NA

SD.2.3.e (CSU)

Are all decision points and transfers of control commented in the CSU?

Y/N/NA

SD.2.4.e (CSU)

Is all machine-dependent code commented in the CSU?

Y/N/NA

SD.2.5.e (CSU)

Are all nonstandard HOL statements commented in the CSU?

Y/N/NA

SD.2.6.e (CSU)

Are the attributes (usage, properties, units of measure) of all declared variables described by comments?

Y/N/NA

SD.2.7.e (CSU)

Do all the comments related to operations in the CSU describe the purpose or intent of the operation? (For example, comment states "INCREMENT TABLE LOOK-UP INDEX", rather than "INCREMENT A BY 1".)

Y/N/NA

SD.2.8.e (CSU)

Are the range of values and default conditions associated with all input parameters described by comment?

Y/N/NA

SD.3.1.e (Procedure)

Is the procedure coded using only a higher order language?

Y/N/NA

SD.3.2.e (CSU)

Are all variable names in the CSU descriptive of the physical or functional property they represent? (For example, variable names "XCOORD, YCOORD" rather than "A1, A2".)

Y/N/NA

SD.3.3.e (CSU)
Is all the code in the CSU logically blocked and indented?
Y/N/NA

SD.3.4.e (Procedure)
How many lines of code are there in the procedure with more than one statement?
__/NA

SD.3.5.e (Procedure)
How many continuation lines of code are there in the procedure?
__/NA

SD.3.7.e (CSU)
Is the CSU structured in the standard established format?
Y/N/NA

SD.3.8.e (Procedure)
Are all language keywords in the procedure used only with their predefined meaning? (For example, no keywords are also used as variable names.)
Y/N/NA

SI.1.2.e (Procedure)
Is the procedure independent of the source of input and destination of output?
Y/N/NA

SI.1.3.e (CSU)
Is the CSU independent of knowledge of prior processing?
Y/N/NA

SI.1.4.e (CSU)
Does the CSU description/prologue include input, output, processing, and limitations?
Y/N/NA

SI.1.5.e (Procedure)
How many non-tasking entrances into the procedure?
__/NA

SI.1.6.e (Procedure)
How many non-tasking exits from the procedure?
__/NA

SI.1.7.e (CSCI)
How many unique data items are in common blocks in this CSCI?

__/NA

SI.1.8.e (CSCI)
How many unique common blocks in this CSCI?

__/NA

SI.1.11.e (CSU)
Does this CSU description identify all interfacing CSUs and interfacing hardware?

Y/N/NA

SI.3.1.e (Procedure)
How many conditional branch statements are there in the procedure? (For example, IF, WHILE, REPEAT, DO/FOR loop, CASE.)

__/NA

SI.3.2.e (Procedure)
How many unconditional branch statements are there in the procedure? (For example, GOTO, CALL, RETURN.)

__/NA

SI.4.1.e (Procedure)
Is the procedure's flow of control from top to bottom (i.e., control does not erratically jump)?

Y/N/NA

SI.4.3.e (Procedure)
How many negative boolean and compound boolean expressions are used in the procedure?

__/NA

SI.4.5.e (Procedure)
How many loops have unnatural exits? (For example, gotos exiting out of loops, return statements.)

__/NA

SI.4.6.e (Procedure)
How many iteration loops are used in the procedure (DO/FOR loops)?

__/NA

SI.4.7.e (Procedure)

In how many iteration loops are indices modified to alter the fundamental processing of the loop? (Guaranteed value of 0 for Ada.)

__/NA

SI.4.8.e (CSU)

Is the CSU free from all self modification of code? (For example, the unit doesn't alter instructions, overlays of code, etc.)

Y/N/NA

SI.4.9.e (Procedure)

How many statement labels are used in the procedure, excluding labels for format statements?

__/NA

SI.4.10.e (Procedure)

What is the maximum nesting level in the procedure?

__/NA

SI.4.11.e (Procedure)

How many total branches (conditional and unconditional) are used in the procedure?

__/NA

SI.4.12.e (Procedure)

How many data declaration statements are there in the procedure?

__/NA

SI.4.13.e (Procedure)

How many data manipulation statements are there in the procedure?

__/NA

SI.4.14.e (Procedure)

How many total data items (local and global) are used in the procedure?

__/NA

SI.4.15.e (Procedure)

How many local data items are referenced locally in the procedure? (For example, variables declared locally and value parameters.)

__/NA

SI.4.16.e (CSU)

Does each data item in the CSU have a single use? (For example, each array serves only one purpose.)

Y/N/NA

SI.4.17.e (CSU)
Is this CSU coded according to the required programming standard?
Y/N/NA

SI.4.18.e (CSCI)
Is repeated and redundant code avoided by utilizing procedures, functions, generics, tasking and packages?
Y/N/NA

SI.5.1.e (Procedure)
How many data items are used as input to the procedure?
__/NA

SI.5.2.e (Procedure)
How many data items are used for output by the procedure?
__/NA

SI.5.3.e (Procedure)
How many parameters in the procedure's calling sequence return output values?
__/NA

SI.5.4.e (Procedure)
Does the procedure perform a single, non-divisible capability?
Y/N/NA

SI.6.1.e (Procedure)
How many unique operators are in the procedure?
__/NA

SI.6.2.e (Procedure)
How many unique operands are in the procedure?
__/NA

SI.6.3.e (Procedure)
How many total operands are in the procedure?
__/NA

ST.1.1.e (Procedure)
How many data items are in this procedure's interface (i.e., data items used to input or output data to the procedure)?
__/NA

ST.1.2.e (CSU)

How many global data items in this CSU's interface are not adequately commented (i.e., lack comments regarding the purpose, type, limitations of the data)?

__/NA

ST.1.3.e (Procedure)

How many interface data items are in the procedure with negative qualification logic? (For example, boolean values that return "true" upon failure rather than success.)

__/NA

ST.1.4.e (CSU)

Is the CSU interface established solely by arguments in the calling sequence parameter list (i.e., is the CSU free from references to global data)?

Y/N/NA

ST.1.5.e (CSU)

Does the CSU modify the internal code or data of other CSUs?

Y/N/NA

ST.2.1.e (Procedure)

How many unique execution paths are in the procedure, including those caused by a "raise" statement?

__/NA

ST.2.2.e (Procedure)

How many conditional branch statements are in the procedure? (For example, IF, WHILE, EXIT WHEN, CASE.)

__/NA

ST.2.3.e (Procedure)

How many other procedures are called by this procedure? (For example, calls to other functions, generics, procedures, and tasks.)

__/NA

ST.2.4.e (Procedure)

How many iteration loops are there in the procedure? (For example, DO/FOR loops.)

__/NA

ST.2.5.e (CSU)

Are there comments regarding the CSUs called by this CSU and the CSUs which call this CSU?

Y/N/NA

ST.3.4.e (CSU)

Is temporary storage (work space reserved for intermediate or partial results) used only by this CSU during execution (i.e., not shared with other CSUs)?

Y/N/NA

ST.3.5.e (CSU)

Does the CSU mix the management of primary and secondary storage resources with the management of data within the storage areas? (For example, an executive CSU that allocates storage for process, and controls what data can be accessed during process execution?)

Y/N/NA

ST.4.3.e (CSCI)

How many global data items are referenced in the CSCI? (a global variable is defined as a variable declared or visible in a CSCI other than that containing the code referencing the variable).

__/NA

ST.4.4.e (CSCI)

How many global data items are modified (set or changed) by one CSU and referenced by other CSUs? (a global variable is defined as a variable declared or visible in a CSU other than that containing the code referencing the variable).

__/NA

ST.4.5.e (Procedure)

Does this procedure have a single entrance (all procedures calling this procedure must enter at the same location)? (In Ada, all non-tasking units will result in a guaranteed "Y".)

Y/N/NA

ST.4.6.e (CSU

Does this CSU's communication with all interfacing CSUs pass only data parameters (i.e., does not pass any control elements)?

Y/N/NA

ST.5.1.e (CSU)

Is the CSU free from unnecessarily recomputing the same value?

Y/N/NA

ST.5.2.e (CSU)

Is the CSU free from statements which cannot be executed?

Y/N/NA

ST.5.3.e (CSU)

Is the meaning of each data item consistent throughout the CSU (i.e., the use associated with each data item does not change)?

Y/N/NA

ST.5.4.e (CSU)

Is the CSU free from unnecessary intermediate data items?

Y/N/NA

VS.1.1.e (Procedure)

How many execution paths are there in the procedure?

__/NA

VS.1.2.e (Procedure)

How many execution paths are tested in the procedure?

__/NA

VS.1.3.e (Procedure)

How many total input parameters are there in the procedure?

__/NA

VS.1.4.e (Procedure)

How many input parameters are tested in the procedure?

__/NA

VS.3.2.e (CSC)

Are all CSUs in the CSC to be exercised during CSC testing?

Y/N/NA

VS.3.3.e (CSU)

Is there a summary table listing all inputs and outputs for testing?

Y/N/NA

DATA COLLECTION FORM -- F-LEVEL

AC.1.5.f (CSC)

During execution, are all outputs within the specified accuracy tolerances?

Y/N/NA

AM.1.3.f (CSC)

How many error conditions are identified?

__/NA

AM.1.4.f (CSC)

How many recognized error conditions require recovery or repair of the error?

__/NA

AM.1.5.f (CSC)

When an error condition is detected, is its resolution determined by the calling CSC?

Y/N/NA

AM.2.3.f (CSC)

Are all applicable external inputs checked with respect to specified ranges before use?

Y/N/NA

AM.2.4.f (CSC)

Are all applicable external inputs checked with respect to illegal combinations and conflicting requests prior to use?

Y/N/NA

AM.2.5.f (CSC)

Are all detected errors with respect to applicable external inputs reported before processing begins?

Y/N/NA

AM.2.6.f (CSC)

Is there a check to see if all data is available before processing begins?

Y/N/NA

AM.3.1.f (CSC)

Is recovery provided for all computational failures?

Y/N/NA

AM.7.1.f (CSC)

Is recovery made from all failures to communicate with other nodes/systems?.

Y/N/NA

AM.7.2.f (CSC)

Are adjacent nodes or interoperating systems periodically checked for operational status?

Y/N/NA

AM.7.3.f (CSC)

Are there provisions for alternate message routing?

Y/N/NA

AP.1.1.f (CSC)

Are references to the database management scheme limited?

Y/N/NA

AP.3.1.f (CSC)

Are specific references to the computer architecture localized (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.f (CSC)

Is the CSC free from microcode instructions?

Y/N/NA

AP.5.1.f (CSC)

Were algorithms incorporated which were used by other CSC applications? (For example, table-driven algorithms.)

Y/N/NA

AP.5.2.f (CSCI)

How many CSCs are there?

__/NA

AP.5.6.f (SYSTEM)

How many CSCIs are there?

__/NA

AT.1.3.f (CSC)

What is the total memory space allocated to the CSC?

__/NA

AT.1.4.f (CSC)
What is the total memory space used by the CSC?
__/NA

AT.1.6.f (CSC)
What is the total auxiliary space allocated to the CSC?
__/NA

AT.1.7.f (CSC)
What is the total auxiliary space used by the CSC?
__/NA

AT.2.4.f (CSC)
What is the total processing time allocated to the CSC?
__/NA

AT.2.5.f (CSC)
What is the total processing time used by the CSC?
__/NA

AT.3.2.f (CSC)
What is the total I/O channel time allocated to the CSC?
__/NA

AT.3.3.f (CSC)
What is the total I/O channel time used by the CSC?
__/NA

AT.3.5.f (CSC)
What is the total communication channel time allocated to the CSC?
__/NA

AT.3.6.f (CSC)
What is the total communication channel time used by the CSC?
__/NA

CL.2.2.f (CSC)
Do all data representations and translations between representations of data comply with the established standard?
Y/N/NA

CL.2.3.f (CSU)
Does the unit perform data translation?
Y/N/NA

CP.1.12.f (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.f (CSCI)

How many recorded problem reports have been corrected to date?

__/NA

CS.2.14.f (CSC)

Do all references to the same data use single unique names?

Y/N/NA

DO.1.1.f (CSCI)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.f (CSCI)

Is the documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.3.f (CSCI)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document. (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.4.f (CSCI)

Does all the test documentation have separate volumes or separations within a single volume based on CSCI capabilities or software elements?

Y/N/NA

EC.1.1.f (CSC)

Does the CSC meet its specified performance requirements and limitations for system communication efficiency?

Y/N/NA

EP.1.1.f (CSC)

Does the CSC meet its specified performance requirements for processing efficiency?

Y/N/NA

EP.2.1.f (CSCI)

Does the CSCI meet its performance requirements and limitations for storing data?

Y/N/NA

GE.2.3.f (CSC)

Is this CSC free from machine-dependent operations (e.g., no representation clauses, compiler predefined pragmas, or use of system-defined types)?

Y/N/NA

GE.2.4.f (CSC)

Can the volume of data processed by the CSC be changed without effecting the executable code? (For example, data volume limits are parameterized.)

Y/N/NA

GE.2.5.f (CSC)

Can the range of data input be changed without effecting the executable code? (For example, no error tolerances are specified, no range-tests or reasonableness checks are performed.)

Y/N/NA

MO.1.2.f (CSC)

Is the CSC tested according to structured techniques?

Y/N/NA

SY.3.2.f (CSCI)

Does this CSCI use the same interrupt structure as the interoperating system(s) in accordance with requirements?

Y/N/NA

VS.1.3.f (CSC)

How many total input parameters are there in the CSC?

__/NA

VS.1.4.f (CSC)

How many input parameters are tested in the CSC?

__/NA

VS.2.1.f (CSC)

How many total interfaces are there between CSUs in the CSC?

__/NA

VS.2.2.f (CSC)

How many CSU interfaces are tested?

__/NA

VS.3.2.f (Procedure)

Is the procedure exercised during CSCI testing?

Y/N/NA

VS.3.3.f (CSC)

Is there a summary table listing all inputs and outputs for testing?

Y/N/NA

VS.4.1.f (CSC)

Are CSC capability requirements tested?

Y/N/NA

DATA COLLECTION FORM -- G-LEVEL

AC.1.5.g (CSCI)

During execution, are all outputs within the specified accuracy tolerances?

Y/N/NA

AM.1.3.g (CSCI)

How many error conditions are identified?

__/NA

AM.1.4.g (CSCI)

How many identified error conditions are provided with processing instructions for recovery or repair of the error?

__/NA

AM.1.5.g (CSCI)

When an error condition is detected, is its resolution determined by the calling body?

Y/N/NA

AM.3.1.g (CSCI)

Are computational failures detected and recovery made?

Y/N/NA

AM.3.2.g (CSCI)

Are all critical (i.e., supporting a mission-critical capability) loop and index parameters checked by explicit checks in the code or by features of the Ada language for out-of-range values before use?

Y/N/NA

AM.3.3.g (CSCI)

Are all critical (i.e., supporting a mission-critical capability) subscript values checked for out-of-range values before use?

Y/N/NA

AM.3.4.g (CSCI)

Are all critical (i.e., supporting a mission-critical capability) outputs checked for reasonable values before final outputting?

Y/N/NA

AM.4.1.g (CSCI)

Is recovery made (e.g., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, hardware failure, clock interrupt)?

Y/N/NA

AM.5.1.g (CSCI)
Does the CSCI recover from all I/O device errors (i.e., I/O Exceptions - Device Error)?
Y/N/NA

AM.6.1.g (CSCI)
Does the CSCI recover from all communications transmission errors?
Y/N/NA

AM.6.2.g (CSCI)
Are all messages transmitted with error checking information? (For example, checksum, parity bit.)
Y/N/NA

AM.6.3.g (CSCI)
Is error checking information recomputed and compared upon receipt of all messages?
Y/N/NA

AM.6.4.g (CSCI)
Are transmission retries limited for all messages?
Y/N/NA

AM.7.1.g (CSCI)
Is recovery made from all failures to communicate with other nodes/systems?
Y/N/NA

AM.7.2.g (CSCI)
Are adjacent nodes or interoperating systems periodically checked for operational status?
Y/N/NA

AM.7.3.g (CSCI)
Are there provisions for alternate message routing?
Y/N/NA

AP.1.1.g (CSCI)
Is the CSCI free from references to the database management scheme (e.g., all data calls for data base information are processed through an executive)?
Y/N/NA

AP.3.1.g (CSCI)

Are specific references to the computer architecture localized (for example, specific device references localized to the executive rather than application software), including representation clauses, package system, and unchecked conversions?

Y/N/NA

AP.4.1.g (CSCI)

Are microcode instruction statements avoided or limited?

Y/N/NA

AP.5.1.g (CSCI)

Are algorithms incorporated which are not unique to this CSCI's application? (For example, table-driven algorithms.)

Y/N/NA

AP.5.6.g (SYSTEM)

How many CSCIs are there?

__/NA

AT.1.3.g (CSCI)

What is the total memory space allocated to the CSCI?

__/NA

AT.1.4.g (CSCI)

What is the total memory space used by the CSCI?

__/NA

AT.1.6.g (CSCI)

What is the total auxiliary space allocated to the CSCI?

__/NA

AT.1.7.g (CSCI)

What is the total auxiliary space used by the CSCI?

__/NA

AT.2.4.g (CSCI)

What is the total processing time allocated to the CSCI?

__/NA

AT.2.5.g (CSCI)

What is the total processing time used by the CSCI?

__/NA

AT.3.2.g (CSCI)

What is the total I/O channel time allocated to the CSCI?

__/NA

AT.3.3.g (CSCI)

What is the total I/O channel time used by the CSCI?

__/NA

AT.3.5.g (CSCI)

What is the total communication channel time allocated by the CSCI?

__/NA

AT.3.6.g (CSCI)

What is the total communication channel time used by the CSCI?

__/NA

AT.4.1.g (CSCI)

Is there compatibility among all processors, communication links, memory devices, and peripherals within the CSCI?

Y/N/NA

AU.1.6.g (CSCI)

What is the CPU processing time in seconds for executing hardware/device interface protocol?

__/NA

AU.1.7.g (CSCI)

What is the processing CPU time in seconds for the entire CSCI?

__/NA

AU.2.2.g (CSCI)

Does executive software test its own operation, communication links, memory devices, and peripheral devices?

Y/N/NA

CL.1.1.g (CSCI)

Is there communication with other systems?

Y/N/NA

CL.1.2.g (CSCI)

Do protocol standards control all network communications?

Y/N/NA

CL.1.5.g (CSCI)
Is all network processing controlled?
Y/N/NA

CL.1.6.g (CSCI)
Is all network processing controlled in accordance with the protocol standards?
Y/N/NA

CL.1.7.g (CSCI)
Are use sessions controlled in accordance with the specified requirements?
Y/N/NA

CL.1.8.g (CSCI)
Are all user sessions controlled in accordance with the protocol standards?
Y/N/NA

CL.1.9.g (CSCI)
Is the communications routing strategy in accordance with the requirements?
Y/N/NA

CL.1.10.g (CSCI)
Is the communication routing strategy in accordance with the protocol standards?
Y/N/NA

CL.1.12.g (CSCI)
Is message handling (e.g., synchronization, message decoding) performed in a uniform manner in accordance with the protocol standards?
Y/N/NA

CL.1.17.g (CSCI)
Is there a description of how the computer system will appear to its users and how the users will interact with the system?
Y/N/NA

CL.1.18.g (CSCI)
Is there a complete and definitive set of operating procedures for using this system?
Y/N/NA

CL.2.2.g (CSCI)

Do all data representations and translations between representations of data comply with the established standard?

Y/N/NA

CL.2.3.g (CSCI)

How many capabilities perform data translations?

__/NA

CL.2.5.g (CSCI)

How many different formats are used for input data from other systems (e.g., formats for data positioning, data packing, block transmission)?

__/NA

CL.2.7.g (CSCI)

How many different formats are used to output data to other systems (e.g., formats for data positioning, data packing, block transmission)?

__/NA

CL.2.8.g (CSCI)

How many different types of input records are received from other systems?

__/NA

CL.2.9.g (CSCI)

How many types of input records transmitted from other systems contain tags identifying the type of data they contain?

__/NA

CL.2.10.g (CSCI)

How many different types of output records are transmitted to other systems?

__/NA

CL.2.11.g (CSCI)

How many types of output records to other systems contain tags identifying the type of data they contain?

__/NA

CP.1.12.g (CSCI)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.g (CSCI)

How many recorded problem reports have been corrected to date?

__/NA

CS.2.11.g (CSCI)

Have procedures been implemented to establish consistency and concurrency for multiple copies of the same information?

Y/N/NA

CS.2.13.g (CSCI)

Have procedures been implemented to verify consistency and concurrency between multiple copies of the same information?

Y/N/NA

CS.2.14.g (CSCI)

Do all references to the same data use single unique names?

Y/N/NA

DI.1.3.g (CSCI)

Is information organized and distributed within the CSCI? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.g (CSCI)

Can each node access all files/libraries in accordance with the specified requirements?

Y/N/NA

DI.1.5.g (CSCI)

Can alternate processing sources be selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.6.g (CSCI)

Are all mission-critical capabilities distributed over redundant elements/nodes in order to reduce vulnerability?

Y/N/NA

DI.1.8.g (CSCI)

Are all control capabilities distributed across different nodes/elements to ensure system operation under anomalous conditions?

Y/N/NA

DI.1.9.g (CSCI)

Are CSCI capabilities implemented across several physically separated components that make up the distributed architecture?

Y/N/NA

DI.1.10.g (CSCI)

Can each node communicate with all remaining nodes, in the event a node becomes unavailable, in accordance with the specified requirements?

Y/N/NA

DO.1.1.g (CSCI)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.g (CSCI)

Is the documentation structured such that capabilities are separately specified?

Y/N/NA

DO.2.3.g (CSCI)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document. (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.4.g (CSCI)

Does all the test documentation have separate volumes or separations within a single volume based on CSCI capabilities or software elements?

Y/N/NA

DO.2.8.g (CSCI)

Are all the software listings included in the software documentation?

Y/N/NA

EC.1.1.g (CSCI)

Does the CSCI meet its specified performance requirements and limitations for system communication efficiency?

Y/N/NA

EP.1.1.g (CSCI)

Does the CSCI meet its specified performance requirements for processing efficiency?

Y/N/NA

EP.1.9.g (CSCI)

How many different overlays are used in the CSCI?

__/NA

EP.2.1.g (CSCI)

Does the CSCI meet its performance requirements and limitations for storing data?

Y/N/NA

ES.1.1.g (CSCI)

Does each capability meet the specified performance requirements to efficiently use primary and secondary storage?

Y/N/NA

ES.1.4.g (CSCI)

Was storage efficiency optimized by using an optimizing compiler, the pragma OPTIMIZE, or coding in assembly language?

Y/N/NA

ES.1.11.g (CSCI)

Is the CSCI free from redundant storage of files and libraries? (For example, duplicate copies of files are not stored at different nodes, multiple versions of the same file are not part of the working CSCI.)

Y/N/NA

FS.3.1.g (CSCI)

Can the user choose among computation and output options?

Y/N/NA

FS.3.2.g (CSCI)

Can resources allocated to capabilities be modified during program execution?

Y/N/NA

GE.2.3.g (CSCI)

Is this CSCI free from machine-dependent operations (e.g., no representation clauses, compiler predefined pragmas, or use of system-defined types)?

Y/N/NA

GE.2.4.g (CSCI)

Can the volume of data processed by the unit be changed without effecting the executable code? (For example, data volume limits are parameterized.)

Y/N/NA

GE.2.5.g (CSCI)

Can the range of data input be changed without effecting the executable code? (For example, no error tolerances are specified, no range-tests or reasonableness checks are performed.)

Y/N/NA

MO.1.2.g (CSCI)

Is the CSCI tested according to structured techniques?

Y/N/NA

OP.1.1.g (CSCI)

Has a description of the operating characteristics of the CSCI been provided (i.e., the normal and alternate procedures and actions performed by the CSCI)? (For example, operating characteristics are described in an operator's manual.)

Y/N/NA

OP.1.2.g (CSCI)

Are all specified error conditions reported to the operator/user such that the nature of the error, and any response required by the operator/user are clearly identified and described in the error message?

Y/N/NA

OP.1.3.g (CSCI)

Is the capability provided for operator/user response to all reported errors as specified in the requirements?

Y/N/NA

OP.1.4.g (CSCI)

Can the operator interrupt operation, obtain operational status, save and enter data, and continue processing?

Y/N/NA

OP.1.6.g (CSCI)

Are the procedures specified for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.g (CSCI)

Is a hard copy log provided of all operator interactions with the CSCI?

Y/N/NA

OP.1.8.g (CSCI)

Are all operator messages and required responses simple and consistent (i.e., there are a minimal number of operator message and response formats, and the same format types are used throughout the CSCI)?

Y/N/NA

OP.1.10.g (CSCI)

Are all access violations reported to the operator in accordance with the CSCI requirements?

Y/N/NA

OP.1.11.g (CSCI)

Are appropriate responses performed to all access violations in accordance with the CSCI requirements?

Y/N/NA

OP.1.12.g (CSCI)

Can the CSCI obtain specific system (or network) resource status information and reallocate resources?

Y/N/NA

OP.1.13.g (CSCI)

Can the operator/user select different nodes for different types of processing or for retrieval of different information?

Y/N/NA

OP.1.14.g (CSCI)

Can the operator/user manipulate data regardless of the data's location in the system?

Y/N/NA

OP.1.15.g (CSCI)

Are the system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.1.g (CSCI)

What are the total number of user input parameters to system capabilities?

__/NA

OP.2.2.g (CSCI)

How many user input parameters have default values?

__/NA

OP.2.3.g (CSCI)

How many different input formats must the user be familiar with?

__/NA

OP.2.4.g (CSCI)

How many input parameters can the user "self describe" along with the parameter value? (For example, user inputs: "TARGETS = 2.")

__/NA

OP.2.6.g (CSCI)

Can the user review and modify all input data prior to execution?

Y/N/NA

OP.2.7.g (CSCI)

Is all user input terminated by an explicitly defined logical end of input?

Y/N/NA

OP.2.8.g (CSCI)

Can the user choose among options for input media?

Y/N/NA

OP.3.1.g (CSCI)

Can the user control output?

Y/N/NA

OP.3.2.g (CSCI)

Do all outputs to the user have unique descriptive labels which identify the data?

Y/N/NA

OP.3.3.g (CSCI)

Are all output data items provided in units descriptive to the user?

Y/N/NA

OP.3.4.g (CSCI)

How many different formats are output to the user? (For example, CRT display arrangements, printer outputs.)

__/NA

OP.3.5.g (CSCI)

Are all user outputs separated into logical groups to facilitate user examination?

Y/N/NA

OP.3.6.g (CSCI)

Do all error messages clearly identify the nature of the error to the user?

Y/N/NA

OP.3.7.g (CSCI)

Can the user output data to different media?

Y/N/NA

OP.3.8.g (CSCI)

Is there a standard user command language for network information and data access?

Y/N/NA

RE.1.1.g (CSCI)

Do communication paths exist to all remaining nodes/communication links, in the event of failure of one node/link?

Y/N/NA

RE.1.2.g (CSCI)

Is the integrity of all data values maintained, following anomalous conditions?

Y/N/NA

RE.1.3.g (CSCI)

Can all disconnected nodes rejoin the network after recovery, such that the processing capabilities of the system are not interrupted?

Y/N/NA

RE.1.4.g (CSCI)

Are all critical data in the CSCI replicated at two or more distinct nodes, in accordance with specified requirements?

Y/N/NA

SI.2.1.g (CSCI)

Does the CSCI use a structure language or preprocessor in accordance with requirements?

Y/N/NA

SS.1.1.g (CSCI)

Are there controls on user input/output access, in the CSCI in accordance with the specified requirements? (For example, user access is limited by identification and password checking.)

Y/N/NA

SS.1.2.g (CSCI)

Is data access controlled in the CSCI in accordance with specified requirements? (For example, authorization tables and privacy locks control data access.)

Y/N/NA

SS.1.3.g (CSCI)

Is the scope of task operations during execution controlled, in accordance with the specified requirements? (For example, tasks cannot invoke other tasks, access system registers, or use privileged commands.)

Y/N/NA

SS.1.4.g (CSCI)

Is access to the network controlled, in accordance with the specified requirements?

Y/N/NA

SS.2.1.g (CSCI)

Is all access to the system recorded and reported, in accordance with the specified requirements? (For example, terminal and processor linkage, data file access, and jobs-run information are recorded.)

Y/N/NA

SS.2.2.g (CSCI)

Are all access violations immediately indicated and identified, in accordance with the specified requirements?

Y/N/NA

SY.1.1.g (CSCI)

Does this CSCI use the same system I/O transmission rate as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.2.g (CSCI)

Does this CSCI use the same communication protocol as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.3.g (CSCI)

Is there a common interpretation of the content of all messages sent from and received by this CSCI and the interoperating system(s), in accordance with the specified requirements? (For example, all variables in the message have the same meaning.)

Y/N/NA

SY.1.4.g (CSCI)

Does this CSCI use the same structure and sequence for message contents as the interoperating system(s), in accordance with the specified requirements? (For example, all real variables are 16 bits in length, and real coordinates are ordered Xcoord, Ycoord, Zcoord.)

Y/N/NA

SY.2.1.g (CSCI)

Does this CSCI use the same data format as the interoperating system(s), in accordance with the specified requirements? (For example, all characters are represented in ASCII format.)

Y/N/NA

SY.2.2.g (CSCI)

Has this CSCI established the same data base structure as the interoperating system(s), in accordance with the specified requirements? (For example, all systems use a relational data base containing similar information.)

Y/N/NA

SY.2.3.g (CSCI)

Does this CSCI provide the same data base access techniques as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.3.1.g (CSCI)

Does this CSCI use the same word length as the interoperating system in accordance with requirements?

Y/N/NA

SY.3.2.g (CSCI)

Does this CSCI use the same interrupt structure as the interoperating system(s) in accordance with requirements?

Y/N/NA

SY.3.3.g (CSCI)

Does this CSCI use the same instruction set as the interoperating system(s)?

Y/N/NA

SY.4.1.g (CSCI)

Does this CSCI use the same source code language(s) as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.4.2.g (CSCI)

Does this CSCI use the same operating system as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.4.3.g (CSCI)

Does this CSCI use the same support software as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

TN.1.1.g (CSCI)

Have lesson plans and training materials been provided for operators, end users, and maintainers of the CSCI?

Y/N/NA

TN.1.2.g (CSCI)

Have realistic simulation exercises for the CSCI been provided?

Y/N/NA

TN.1.3.g (CSCI)

Are "HELP" information and diagnostic information provided for the operator, end user, and maintainer of the CSCI? (For example, an on-line list of legal commands or a list of the sequential steps in a process are provided.)

Y/N/NA

TN.1.4.g (CSCI)

Can CSCI users select the level of aid and guidance, according to their degree of expertise?

Y/N/NA

VS.1.3.g (CSCI)

How many total input parameters are there?

__/NA

VS.1.4.g (CSCI)

How many input parameters are to be tested?

__/NA

VS.2.2.g (CSCI)

Are all specified performance requirements of the CSCI tested?

Y/N/NA

VS.3.1.g (CSCI)

Are all specified performance requirements of the CSCI tested?

Y/N/NA

VS.3.2.g (CSCI)

Are all CSUs of the CSCI exercised during CSCI testing?

Y/N/NA

VS.3.3.g (CSCI)

Is there a summary table listing all test inputs and outputs for the CSCI?

Y/N/NA

VS.4.1.g (CSCI)

Are CSCI capability requirements tested?

Y/N/NA .

DATA COLLECTION FORM -- H-LEVEL

AC.1.5.h (SYSTEM)

During execution, are all outputs within the specified accuracy tolerances?

Y/N/NA

AM.1.3.h (SYSTEM)

How many error conditions are identified?

__/NA

AM.1.4.h (SYSTEM)

How many identified error conditions are provided with processing instructions for recovery or repair of the error?

__/NA

AM.1.5.h (SYSTEM)

When an error condition is detected, is its resolution determined by the calling body?

Y/N/NA

AM.3.1.h (SYSTEM)

Are computational failures detected and recovery made?

Y/N/NA

AM.4.1.h (SYSTEM)

Is recovery made (e.g., exception handlers or other means) from all detected hardware faults (e.g., arithmetic faults, hardware failure, clock interrupt)?

Y/N/NA

AM.5.1.h (SYSTEM)

Does the system recover from all I/O device errors (i.e., I/O Exceptions - Device Error)?

Y/N/NA

AM.6.1.h (SYSTEM)

Does the system recover from all communications transmission errors?

Y/N/NA

AM.7.1.h (SYSTEM)

Is recovery made from all failures to communicate with other nodes/systems?

Y/N/NA

AT.2.5.h (SYSTEM)
What is the total processing time used by the system?

__/NA

AT.3.2.h (SYSTEM)
What is the total I/O channel time allocated to the system?

__/NA

AT.3.3.h (SYSTEM)
What is the total I/O channel time used by the system?

__/NA

AT.3.5.h (SYSTEM)
What is the total communication channel time allocated by the system?

__/NA

AT.3.6.h (SYSTEM)
What is the total communication channel time used by the system?

__/NA

AT.4.1.h (SYSTEM)
Is there compatibility among all processors, communication links, memory devices, and peripherals within the system?

Y/N/NA

AU.2.1.h (SYSTEM)
Does each CPU/SYSTEM have a separate power source?

Y/N/NA

AU.2.2.h (SYSTEM)
Does executive software test its own operation, communication links, memory devices, and peripheral devices?

Y/N/NA

CL.1.1.h (SYSTEM)
Is there communication with other systems?

Y/N/NA

CL.1.2.h (SYSTEM)
Do protocol standards control all network communications?

Y/N/NA

CL.1.5.h (SYSTEM)
Is all network processing controlled?

Y/N/NA

CL.1.6.h (SYSTEM)

Is all network processing controlled in accordance with the protocol standards?

Y/N/NA

CL.1.7.h (SYSTEM)

Are use sessions controlled in accordance with the specified requirements?

Y/N/NA

CL.1.8.h (SYSTEM)

Are all user sessions controlled in accordance with the protocol standards?

Y/N/NA

CL.1.9.h (SYSTEM)

Is the communications routing strategy in accordance with the requirements

Y/N/NA

CL.1.10.h (SYSTEM)

Is the communication routing strategy in accordance with the protocol standards?

Y/N/NA

CL.1.11.h (SYSTEM)

Are messages handled in a uniform manner?

Y/N/NA

CL.1.12.h (SYSTEM)

Is message handling (e.g., handshaking, message decoding) performed in a uniform manner in accordance with the protocol standards?

Y/N/NA

CL.1.17.h (SYSTEM)

Is there a description of how the computer system will appear to its users and how the users will interact with the system?

Y/N/NA

CL.1.18.h (SYSTEM)

Is there a complete and definitive set of operating procedures for using this system?

Y/N/NA

CL.2.2.h (SYSTEM)
Do all data representations and translations between representations of data comply with the established standard?
Y/N/NA

CL.2.3.h (SYSTEM)
How many system capabilities perform data translations?
__/NA

CL.2.4.h (SYSTEM)
Is all input data from other systems received in common formats? (For example, common format for data positioning, data packing, block transmission.)
Y/N/NA

CL.2.5.h (SYSTEM)
How many different formats are used for input data from other systems (e.g., formats for data positioning, data packing, block transmission)?
__/NA

CL.2.6.h (SYSTEM)
Is all data output to other systems in common formats (e.g., formats for data positioning, data packing, block transmission)?
Y/N/NA

CL.2.7.h (SYSTEM)
How many different formats are used to output data to other systems (e.g., formats for data positioning, data packing, block transmission)?
__/NA

CL.2.8.h (SYSTEM)
How many different types of input records are received from other systems?
__/NA

CL.2.9.h (SYSTEM)
How many types of input records transmitted from other systems contain tags identifying the type of data they contain?
__/NA

CL.2.10.h (SYSTEM)
How many different types of output records are transmitted to other systems?
__/NA

CL.2.11.h (SYSTEM)

How many types of output records to other systems contain tags identifying the type of data they contain?

__/NA

CP.1.12.h (SYSTEM)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.h (SYSTEM)

How many recorded problem reports have been corrected to date?

__/NA

CS.2.11.h (SYSTEM)

Have procedures been implemented to establish consistency and concurrency for multiple copies of the same information?

Y/N/NA

CS.2.13.h (SYSTEM)

Have procedures been implemented to verify consistency and concurrency between multiple versions of the same information?

Y/N/NA

CS.2.14.h (SYSTEM)

Do all references to the same data use single unique names?

Y/N/NA

DI.1.3.h (SYSTEM)

Is information organized and distributed within the system? (For example, information is distributed across nodes or among storage devices.)

Y/N/NA

DI.1.4.h (SYSTEM)

Can each node access all files/libraries in accordance with the specified requirements?

Y/N/NA

DI.1.5.h (SYSTEM)

Can alternate processing sources be selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.6.h (SYSTEM)

Are all mission-critical capabilities distributed over redundant elements/nodes in order to reduce vulnerability?

Y/N/NA

DI.1.8.h (SYSTEM)

Are all control capabilities distributed across different nodes/elements to ensure system operation under anomalous conditions?

Y/N/NA

DI.1.9.h (SYSTEM)

Are CSCI capabilities implemented across several physically separated components that make up the distributed architecture?

Y/N/NA

DI.1.10.h (SYSTEM)

Can each node communicate with all remaining nodes, in the event a node becomes unavailable, in accordance with the specified requirements?

Y/N/NA

DO.1.1.h (SYSTEM)

Are current versions of all software documentation related to the project free from access control (i.e., any member of the current project or other projects may access a copy of any document)?

Y/N/NA

DO.2.1.h (SYSTEM)

Is all the system documentation structured such that capabilities are separately identified?

Y/N/NA

DO.2.3.h (SYSTEM)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.4.h (SYSTEM)

Does all the test documentation have separate volumes or separations within a single volume based on CSCI capabilities or software elements?

Y/N/NA

DO.2.8.h (SYSTEM)

Are all the software listings included in the software documentation?

Y/N/NA

EC.1.1.h (SYSTEM)

Does the system meet its specified performance requirements and limitations for system communication efficiency?

Y/N/NA

EP.1.1.h (SYSTEM)

Does the system meet its specified performance requirements for processing efficiency?

Y/N/NA

EP.1.9.h (SYSTEM)

Is the memory management of the system free from requirements for overlays?

Y/N/NA

EP.2.1.h (SYSTEM)

Does the system meet its performance requirements and limitations for storing data?

Y/N/NA

ES.1.1.h (SYSTEM)

Does each capability meet the specified performance requirements to efficiently use primary and secondary storage?

Y/N/NA

ES.1.4.h (SYSTEM)

Was storage efficiency optimized by using an optimizing compiler, the pragma OPTIMIZE, or coding in assembly language?

Y/N/NA

ES.1.11.h (SYSTEM)

Is the system free from redundant storage of files and libraries? (For example, duplicate copies of files are not stored at different nodes, multiple versions of the same file are not part of the working CSCI.)

Y/N/NA

FO.1.3.h (SYSTEM)

How many duplicated capability sets exist between this system and interoperating systems (i.e., the same function is performed in this system and in interoperating systems)?

__/NA

FO.1.4.h (SYSTEM)

In how many instances of duplicated capability sets were duplicated capabilities deleted (leaving responsibility for performing the capability with one system)?

__/NA

FS.3.1.h (SYSTEM)

Can the user choose among computation and output options?

Y/N/NA

FS.3.2.h (SYSTEM)

Can resources allocated to capabilities be modified during program execution?

Y/N/NA

GE.2.3.h (SYSTEM)

Is this system free from machine-dependent operations (e.g., no representation clauses, compiler predefined pragmas, or use of system-defined types)?

Y/N/NA

GE.2.4.h (SYSTEM)

Can the volume of data processed by the unit be changed without effecting the executable code? (For example, data volume limits are parameterized.)

Y/N/NA

GE.2.5.h (SYSTEM)

Can the range of data input be changed without effecting the executable code? (For example, no error tolerances are specified, no range-tests or reasonableness checks are performed.)

Y/N/NA

MO.1.2.h (SYSTEM)

Is the system tested according to structured techniques?

Y/N/NA

OP.1.1.h (SYSTEM)

Has a description of the operating characteristics of the system been provided (i.e., the normal and alternate procedures and actions performed by the CSCI)? (For example, operating characteristics are described in an operator's manual.)

Y/N/NA

OP.1.2.h (SYSTEM)

Are all specified error conditions reported to the operator/user such that the nature of the error, and any response required by the operator/user are clearly identified and described in the error message?

Y/N/NA

OP.1.3.h (SYSTEM)

Is the capability provided for operator/user response to all reported errors as specified in the requirements?

Y/N/NA

OP.1.6.h (SYSTEM)

Are the procedures specified for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.h (SYSTEM)

Is a hard copy log provided of all operator interactions with the system?

Y/N/NA

OP.1.8.h (SYSTEM)

Are all operator messages and required responses simple and consistent (i.e., there are a minimal number of operator message and response formats, and the same format types are used throughout the system)?

Y/N/NA

OP.1.10.h (SYSTEM)

Are all access violations reported to the operator in accordance with the system requirements?

Y/N/NA

OP.1.11.h (SYSTEM)

Are appropriate responses performed to all access violations in accordance with the system requirements?

Y/N/NA

OP.1.12.h (SYSTEM)

Can the user obtain specific system (or network) resource status information and reallocate resources?

Y/N/NA

OP.1.13.h (SYSTEM)

Can the operator/user select different nodes for different types of processing or for retrieval of different information?

Y/N/NA

OP.1.14.h (SYSTEM)

Can the operator/user manipulate data regardless of the data's location in the system?

Y/N/NA

OP.1.15.h (SYSTEM)

Are the system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.1.h (SYSTEM)

What are the total number of user input parameters to system capabilities?

__/NA

OP.2.2.h (SYSTEM)

How many user input parameters have default values?

__/NA

OP.2.3.h (SYSTEM)

How many different input *formats* must the user be familiar with?

__/NA

OP.2.4.h (SYSTEM)

How many input parameters can the user "self describe" along with the parameter value? (For example, user inputs: "TARGETS = 2.")

__/NA

OP.2.6.h (SYSTEM)

Can the user review and modify all input data prior to execution?

Y/N/NA

OP.2.7.h (SYSTEM)

Is all user input terminated by an explicitly defined logical end of input?

Y/N/NA

OP.2.8.h (SYSTEM)

Can the user choose among options for input media?

Y/N/NA

OP.3.1.h (SYSTEM)
Can the user control output?
Y/N/NA

OP.3.2.h (SYSTEM)
Do all outputs to the user have unique descriptive labels which identify the data?
Y/N/NA

OP.3.3.h (SYSTEM)
Are all output data items provided in units descriptive to the user?
Y/N/NA

OP.3.4.h (SYSTEM)
How many different formats are output to the user? (For example, CRT display arrangements, printer outputs.)
__/NA

OP.3.5.h (SYSTEM)
Are all user outputs separated into logical groups to facilitate user examination?
Y/N/NA

OP.3.6.h (SYSTEM)
Do all error messages clearly identify the nature of the error to the user?
Y/N/NA

OP.3.7.h (SYSTEM)
Can the user output data to different media?
Y/N/NA

OP.3.8.h (SYSTEM)
Is there a standard user command language for network information and data access?
Y/N/NA

RE.1.1.h (SYSTEM)
Do communication paths exist to all remaining nodes/communication links, in the event of failure of one node/link?
Y/N/NA

RE.1.2.h (SYSTEM)

Is the integrity of all data values maintained, following anomalous conditions?

Y/N/NA

RE.1.3.h (SYSTEM)

Can all disconnected nodes rejoin the network after recovery, such that the processing capabilities of the system are not interrupted?

Y/N/NA

RE.1.4.h (SYSTEM)

Are all critical data in the CSCI replicated at two or more distinct nodes, in accordance with specified requirements?

Y/N/NA

SI.2.1.h (SYSTEM)

Does the CSCI use a structure language or preprocessor in accordance with requirements?

Y/N/NA

SS.1.1.h (SYSTEM)

Are there controls on user input/output access, in the system in accordance with the specified requirements? (For example, user access is limited by identification and password checking.)

Y/N/NA

SS.1.2.h (SYSTEM)

Is data access controlled in the system in accordance with specified requirements? (For example, authorization tables and privacy locks control data access.)

Y/N/NA

SS.1.3.h (SYSTEM)

Is the scope of task operations during execution controlled, in accordance with the specified requirements? (For example, tasks cannot invoke other tasks, access system registers, or use privileged commands.)

Y/N/NA

SS.1.4.h (SYSTEM)

Is access to the network controlled, in accordance with the specified requirements?

Y/N/NA

SS.2.1.h (SYSTEM)

Is all access to the system recorded and reported, in accordance with the specified requirements? (For example, terminal and processor linkage, data file access, and jobs-run information are recorded.)

Y/N/NA

SS.2.2.h (SYSTEM)

Are all access violations immediately indicated and identified, in accordance with the specified requirements?

Y/N/NA

SY.1.1.h (SYSTEM)

Does this system use the same system I/O transmission rate as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.2.h (SYSTEM)

Does this system use the same communication protocol as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.1.3.h (SYSTEM)

Is there a common interpretation of the *content of all messages sent* from and received by this system and the interoperating system(s), in accordance with the specified requirements? (For example, all variables in the message have the same meaning.)

Y/N/NA

SY.1.4.h (SYSTEM)

Does this system use the same structure and sequence for message contents as the interoperating system(s), in accordance with the specified requirements? (For example, all real variables are 16 bits in length, and real coordinates are ordered Xcoord, Ycoord, Zcoord.)

Y/N/NA

SY.2.1.h (SYSTEM)

Does this system use the same data format as the interoperating system(s), in accordance with the specified requirements? (For example, all characters are represented in ASCII format.)

Y/N/NA

SY.2.2.h (SYSTEM)

Has this system established the same data base structure as the interoperating system(s), in accordance with the specified requirements? (For example, all systems use a relational data base containing similar information.)

Y/N/NA

SY.2.3.h (SYSTEM)

Does this system provide the same data base access techniques as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.3.1.h (SYSTEM)

Does this system use the same word length as the interoperating system in accordance with requirements?

Y/N/NA

SY.3.2.h (SYSTEM)

Does this system use the same interrupt structure as the interoperating system(s) in accordance with requirements?

Y/N/NA

SY.3.3.h (SYSTEM)

Does the system use the same instruction set as the interoperating systems?

Y/N/NA

SY.4.1.h (SYSTEM)

Does the system use the same source code language(s) as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.4.2.h (SYSTEM)

Does the system use the same operating system as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

SY.4.3.h (SYSTEM)

Does the system use the same support software as the interoperating system(s), in accordance with the specified requirements?

Y/N/NA

TN.1.1.h (SYSTEM)

Have lesson plans and training materials been provided for operators, end users, and maintainers of the system?

Y/N/NA

TN.1.2.h (SYSTEM)

Have realistic simulation exercises been provided?

Y/N/NA

TN.1.3.h (SYSTEM)

Are "HELP" information and diagnostic information provided for the operator, end user, and maintainer of the system? (For example, an on-line list of legal commands or a list of the sequential steps in a process are provided.)

Y/N/NA

TN.1.4.h (SYSTEM)

Can users select the level of aid and guidance according to their degree of expertise?

Y/N/NA

VS.1.3.h (SYSTEM)

How many total input parameters are there?

__/NA

VS.1.4.h (SYSTEM)

How many input parameters are tested?

__/NA

VS.2.2.h (SYSTEM)

How many CSCI interfaces are tested?

__/NA

VS.3.1.h (SYSTEM)

Are all specified performance requirements tested?

Y/N/NA

VS.3.2.h (SYSTEM)

Are all CSCIs exercised during testing?

Y/N/NA

VS.3.3.h (SYSTEM)

Is there a summary table listing all test inputs and outputs of the system?

Y/N/NA

VS.4.1.h (SYSTEM)

Are all system capabilities tested?

Y/N/NA

DATA COLLECTION FORM -- I-LEVEL

AC.1.5.i (SYSTEM)

What percentage of system outputs have been within the specified accuracy tolerances?

__/NA

AM.3.1.i (SYSTEM)

For what percentage of computational failures were the failures detected and recovery made?

__/NA

AM.4.1.i (SYSTEM)

For what percentage of all detected hardware faults (e.g., arithmetic faults, hardware failure, clock interrupt) was recovery successfully made?

__/NA

AM.5.1.i (SYSTEM)

For what percentage of all I/O device errors (i.e., I/O Exceptions - Device Error) was recovery made?

__/NA

AM.6.1.i (SYSTEM)

For what percentage of communications transmission errors was recovery successful?

__/NA

AM.7.1.i (SYSTEM)

For what percentage of failures to communicate with other nodes/systems was recovery made?

__/NA

AM.7.3.i (SYSTEM)

Are there provisions for alternate message routing?

Y/N/NA

AP.5.1.i (SYSTEM)

Have any of this application's algorithms been used for other systems under development?

Y/N/NA

AP.5.2.i (SYSTEM)

How many system capabilities are there?

__/NA

CL.1.17.i (SYSTEM)

Is there a description of how the computer system will appear to its users and how the users will interact with the system?

Y/N/NA

CL.1.18.i (SYSTEM)

Is there a complete and definitive set of operating procedures for using this system?

Y/N/NA

CP.1.12.i (SYSTEM)

How many approved problem reports have been recorded to date?

__/NA

CP.1.13.i (SYSTEM)

How many recorded problem reports have been corrected to date?

__/NA

CS.2.11.i (SYSTEM)

Have consistency and concurrency for multiple copies of the same information been maintained?

Y/N/NA

DI.1.5.i (SYSTEM)

Have alternate processing sources been selected within the system? (For example, multiple processors, alternate nodes.)

Y/N/NA

DI.1.10.i (SYSTEM)

In what percentage of the failures of one node have the other nodes still been able to communicate with each other?

__/NA

DO.2.1.i (SYSTEM)

Is the system requirements documentation structured such that procedures, capabilities, algorithms, etc. are separately specified?

Y/N/NA

DO.2.2.i (SYSTEM)

Does the maintenance documentation depict control and data flow (e.g., graphic portrayal with accompanying explanation, PDL)?

Y/N/NA

DO.2.3.i (SYSTEM)

Does each document contain a scheme which facilitates quickly locating and accessing various information in the document? (For example, hierarchical structured table of contents, inserted tabs, index.)

Y/N/NA

DO.2.5.i (SYSTEM)

Does the system requirements documentation characterize the operational capabilities of the software? (For example, identify performance parameters and limitations.)

Y/N/NA

DO.2.8.i (SYSTEM)

Are all the software listings included in the software documentation?

Y/N/NA

FS.2.7.i (SYSTEM)

How many system capabilities were used to satisfy the requirements of other similar applications?

__/NA

FS.3.1.i (SYSTEM)

Do the users choose among computation and output options?

Y/N/NA

FS.3.2.i (SYSTEM)

Are resources allocated to capabilities modified during program execution?

Y/N/NA

OP.1.1.i (SYSTEM)

Has a description of the operating characteristics of the system been provided (i.e., the normal and alternate procedures and actions performed by the CSCI)? (For example, operating characteristics are described in an operator's manual.)

Y/N/NA

OP.1.2.i (SYSTEM)

Are all specified error conditions reported to the operator/user such that the nature of the error, and any response required by the operator/user are clearly identified and described in the error message?

Y/N/NA

OP.1.3.i (SYSTEM)

Is the capability provided for operator/user response to all reported errors as specified in the requirements?

Y/N/NA

OP.1.5.i (SYSTEM)

How many operations/responses are performed by the operator for a typical mission/job?

__/NA

OP.1.6.i (SYSTEM)

Are the procedures specified for setting up a mission/job and completing it?

Y/N/NA

OP.1.7.i (SYSTEM)

Is a hard copy log provided of all operator interactions with the system?

Y/N/NA

OP.1.8.i (SYSTEM)

Are all operator messages and required responses simple and consistent (i.e., there are a minimal number of operator message and response formats, and the same format types are used throughout the system)?

Y/N/NA

OP.1.10.i (SYSTEM)

Are all access violations reported to the operator in accordance with the system requirements?

Y/N/NA

OP.1.11.i (SYSTEM)

Are appropriate responses performed to all access violations in accordance with the system requirements?

Y/N/NA

OP.1.12.i (SYSTEM)

Can the user obtain specific system (or network) resource status information and reallocate resources?

Y/N/NA

OP.1.13.i (SYSTEM)

Can the operator/user select different nodes for different types of processing or for retrieval of different information?

Y/N/NA

OP.1.14.i (SYSTEM)

Can the operator/user manipulate data regardless of the data's location in the system?

Y/N/NA

OP.1.15.i (SYSTEM)

Are the system implementation details transparent to the user? (For example, the user can access a file without knowing its location in the system/network.)

Y/N/NA

OP.2.1.i (SYSTEM)

What are the total number of user input parameters to system capabilities?

__/NA

OP.2.2.i (SYSTEM)

How many user input parameters have default values?

__/NA

OP.2.3.i (SYSTEM)

How many different input formats must the user be familiar with?

__/NA

OP.2.4.i (SYSTEM)

How many input parameters can the user "self describe" along with the parameter value? (For example, user inputs: "TARGETS = 2.")

__/NA

OP.2.6.i (SYSTEM)

Can the user review and modify all input data prior to execution?

Y/N/NA

OP.2.7.i (SYSTEM)

Is all user input data terminated by an explicitly defined logical end of input?

Y/N/NA

OP.2.8.i (SYSTEM)

Can the user choose among options for input media?

Y/N/NA

OP.3.1.i (SYSTEM)

Can the user control output?

Y/N/NA

OP.3.2.i (SYSTEM)

Do all outputs to the user have unique descriptive labels which identify the data?

Y/N/NA

OP.3.3.i (SYSTEM)

Are all output data items provided in units descriptive to the user?

Y/N/NA

OP.3.4.i (SYSTEM)

How many different formats are output to the user? (For example, CRT display arrangements, printer outputs.)

__/NA

OP.3.5.i (SYSTEM)

Are all user outputs separated into logical groups to facilitate user examination?

Y/N/NA

OP.3.6.i (SYSTEM)

Do all error messages clearly identify the nature of the error to the user?

Y/N/NA

OP.3.7.i (SYSTEM)

Can the user output data to different media?

Y/N/NA

OP.3.8.i (SYSTEM)

Is there a standard user command language for network information and data access?

Y/N/NA

RE.1.1.i (SYSTEM)

Have alternate communication paths been used to all remaining nodes/communication links after the failure of one node/link?

Y/N/NA

RE.1.2.i (SYSTEM)

Was the integrity of all data values maintained, following anomalous conditions?

Y/N/NA

RE.1.3.i (SYSTEM)

Did all disconnected nodes rejoin the network after recovery, such that the processing capabilities of the system are not interrupted?

Y/N/NA

SS.2.2.i (SYSTEM)

Have access violations been immediately indicated and identified, in accordance with the specified requirements?

Y/N/NA

3.0 SCORING

Scoring the Software Quality Framework is the process of combining metric element results into increasingly higher levels of abstraction. Rather than looking individually at over 1400 metric element data items applied repeatedly for the various software phases and components, the scoring process allows the user to abstract and combine these results into meaningful calculations. These scoring results support differing views of the system under analysis.

The scoring of metric elements can be accomplished using the following two methodologies:

1. An adherence measurement model. In this method, a score corresponds to the value for the number of opportunities taken for metric compliance related to the number of possible opportunities. This applies to "adherence-oriented" metrics, and is not used for all metric elements. An example of such a score is 2,500 / 5,000. In this model, numbers are not reduced to a ratio (in this case, for example, 1/2).

2. An aggregation measurement model. In this method, aggregations of metric scores are made across the following levels:

   - Scoring across the various elements of the metric framework (i.e., metric elements, metrics, criteria, and factors).

   - Scoring across the architecture of the system (computer software units, computer software components, computer software configuration items, and the system itself).

   - Scoring across the functionality of the system, in the form of quality evaluation results for the system-level capabilities.

Each methodology is described below.

3.1 Adherence Model Scoring

This scoring methodology counts compliances for each of the "adherence-oriented" metrics, while also counting the opportunities that the compliance might have occurred. An example of this is metric element MO.1.9.e. This element concerns control being returned to the calling computer software unit when execution is completed. To score for a CSCI using the adherence model, the number of units for which control is returned to the calling unit when execution is complete are counted, as are the total number of units in the CSCI. The value scored for the metric consists of those two raw data items (for example, 235 out of 891). When scoring to higher levels (for example, combining CSCI scoring to calculate system-level scores), these raw numbers are retained and summed. They are not reduced to ratios, because a compliance of 5,000 out of 10,000 possibilities is much different than a compliance of 1 out of 2 possibilities.

3.2 Aggregation Scoring Model

3.2.1 Framework Composition Scoring

The Software Quality Framework is composed of a hierarchy. At the bottom of the hierarchy are the metric element questions (described in the Data Collection Forms of section 2). These questions are data items that apply directly to system documentation and code. The bottom level of the hierarchy, these *metric elements* are components of software *metrics*. Metrics are combinations of data used to evaluate the system based on specific software concepts (e.g., SS.1 is a metric concerned with access control). Metrics, in turn, compose software *criteria*. Criteria are software-oriented concepts such as System Accessibility. The criteria together form software quality *factors*. Factors are user-oriented concepts such as *Reliability* and *Maintainability*.

It is the nature of the Software Quality Framework that data is collected directly only for the software metric elements. All other levels of the framework (metrics, criteria, and factors) are computed based on a methodology of scoring.

Continued research will enable the metric community to derive recommended scoring methodologies to allow the most effective and meaningful combination of metric elements, metrics, and criteria into higher level results. At this point in the research, however, the scoring methodology is intended only to allow use of the framework in the most effective manner known while providing great flexibility for the users.

To this end, the following methodology for scoring the framework items may be used:

1. The value for a metric consists of a calculation based on the applicable metric elements. For the simplest case, this calculation is a weighted averaging of the metric elements.

2. The value calculated for a criterion consists of the weighted averaging of each metric applicable to it.

3. The value calculated for a factor consists of the weighted averaging of each criteria applicable to it.

Each of these aggregations is explained in more detail below. This methodology treats each metric element, metric, criterion, and factor as a separate entity for each of the development phases and software architecture levels of application. Scores across development phases are not combined within the current framework. Combining scores across levels of application (e.g., computer software unit versus the system level) is discussed in section 3.2.2, and builds on the material of this section.

3.2.1.1 Metric Calculation

Metric elements are the various individual items in the Data Collection Forms of section 2.0. Each element is applied to a particular product of a phase in the software development life cycle and at a particular application level. As an example, element AC.1.1.a (concerning error analysis) is applied at the system level during the requirements phase. Element EP.2.4.e (concerning arithmetic expressions with different sized components) is applied to each unit during the code and integration phase.

To score the parent metric value for these particular metric elements, a weighted average of all applicable elements is used. Weighted, in this context, means that one metric element may contribute more to the calculated metric than other metric element(s). As an example, consider the metric AC.1 (Accuracy), applied at the system level during the requirements phase.

Data Collection Form (DCF) A, in section 2, shows that there are six different metric elements that contribute to the metric AC.1 at this phase and level. These elements are AC.1.1.a, AC.1.2.a, AC.1.4.a, AC.1.5.a, AC.1.6.a, and AC.1.7.a. The calculation of the value for metric AC.1 would follow these steps:

1. For each *Y* answer in a metric element, score that element as 1.

2. For each *N* answer in a metric element, score that element as 0.

3. Calculate the metric score using the following equation:

AC.1 = a*(AC.1.1a) + b*(AC.1.2.a) + c*(AC.1.4.a)
+ d*(AC.1.5.a) + e*(AC.1.6.a) + f*(AC.1.7.a.)

where:

* indicates performing a multiplication

(AA.n.n.a) represents the value of the particular metric element

a, b, c, d, e, and f are multiplying coefficients which sum to the value 1. In this case, with no weighting performed, each coefficient will equal 1/6. Weighting can change the relative values, but the sum must always remain 1.

For any metric element that is not applicable (scored as*N/A*), the coefficient is set to a value of 0. The default value for each coefficient is such as to produce an average (in this case, it is 1/6).

Earlier versions of the Software Quality Framework contained metric elements that were each stand-alone, in the sense that each element was a complete calculation unto itself. The Data Collection Forms of section 2.0, however, are not stand-alone in that sense. To reduce the duplication of data collection and calculation, each item to be collected has been separated out. This means that, on occasion, the metric calculation becomes more complicated than a simple summing of each of the metric elements.

Metric AP.5 (Functional Independence) is an example of this type of metric and data. AP.5, at the system level for the requirements phase, consists of 3 metric elements. These elements are AP.5.1.a, AP.5.2.a, and AP.5.3.a. Each metric element concerns, respectively, a requirement to develop non-unique algorithms, the number of system capabilities, and the number of capabilities with algorithms which are non-unique.

The answer to AP.5.1.a is a *Y/N* type answer, with a resultant score of either *0, 1,* or *N/A*. Both AP.5.2.a and AP.5.3.a, however, receive numeric scores ranging from *N/A* or *0* to the total number of system-level capabilities. Since metric, criteria, and factor scores all range in value from 0 to 1, it is clear that AP.5.2.a and AP.5.3.a do not get scored in the same way as other elements. Both elements together constitute a value for the *ratio of system-level capabilities which contain algorithms that are not unique to the system's application compared to the total number of system-level capabilities.* This means that the metric score for these elements is given as follows:

1. If the answer to element AP.5.1.a is *Y*, score the element as *1*. If the value is *N*, score the element as *0*.

2. Calculate the score for metric AP.5 using the following equation:

   AP.5 = a * (AP.5.1.a) + b * (AP.5.3.a / AP.5.2.a)

   where:

   / indicates performing a division

   (AA.n.n.a) is the value for the respective metric element

   a, b are coefficients summing to 1. If the answer to AP.5.1.a is *N/A*, then a = 0. If the answer to either AP.5.2.a or AP.5.3.a is *N/A*, then b = 0. If all three metric elements are *N/A*, then the metric AP.5 is *N/A*.

Generalized, the methodology for scoring metric elements to metrics for a particular phase and level of application is as follows:

1. Score each *Y/N* question with a 1 for a *Y* answer, and a 0 for a *N* answer.

2. Use the metric formulas to calculate the metric score. Appendix A contains the full text of the formulas. The coefficients shown may default to equal values, or may be weighted if desired. For any element with an answer of *N/A*, the coefficient for that element must be 0.

The user can change the coefficients from the default values whenever a particular metric element is considered to be more important than other metric element(s).

3.2.1.2 Criterion Calculation

Each Software Quality Framework criterion is composed of metrics. The score for each criterion is calculated with a weighted average of the score for all the applicable metrics. For example, at the system level for the requirements phase, the criterion Commonality (CL) is composed of three metrics: CL.1 (Communications Commonality), CL.2 (Data Commonality), and CL.3 (Common Vocabulary). To obtain a score for the criterion at that level, the following calculation is performed:

CL = a * (CL.1) + b * (CL.2) + c * (CL.3)

where:

* indicates performing a multiplication

(AA.n) is the value for the respective metric

a, b, and c are coefficients summing to 1. If the score for any metric is N/A, then the coefficient for that metric is 0.

The default value for the coefficients, in this case, is 1/3. These values may be changed if desired. In general, the user can change these defaults whenever a particular metric is more important in the application than are the other metrics.

Appendix B contains the a complete listing of the criterion formulas.

3.2.1.3 Factor Calculation

Just as criteria are made up of metrics, so are factors made up of criteria. The *score for each factor* is calculated as a weighted average of the scores for all the applicable criteria. For example, at the system level for the requirements phase, the factor *Correctness* is composed of three criteria: CP (Completeness), CS (Consistency), and TC (Traceability) To obtain a score for the factor at that level, the following calculation is performed:

Correctness = a * (CP) + b * (CS) + c * (TC)

where:

* indicates performing a multiplication

(AA) is the score for a particular criterion

a, b, and c are coefficients summing to 1. If the score for any criterion isN/A, then the coefficient for that criterion is 0.

The default value for the coefficients, in this case, is 1/3. These values may be changed if desired. In general, the user can change these defaults whenever a particular criterion is more important in the application than are the other criteria.

Appendix C contains the full listing of the factor formulas.

3.2.2 Software Architecture Scoring

Section 3.2.1, above, discussed in detail how to calculate metric, criterion, and factor scores for particular phases and for particular levels of application (i.e., the software architecture level). That methodology can be used to calculate results for the system as whole, using metric elements that are only applicable at the system level. Results for particular computer software configuration items (CSCIs), using only CSCI-level questions, could also be obtained. Likewise, results for computer software components (CSCs), computer software units (CSUs), and procedures could be obtained for questions that apply to only each of those levels, respectively.

However, there is a further consideration that needs to be applied in the scoring methodology. The system-level results are not only the answers to those questions that are applied at the system level. They are also concerned with questions applied to the lower levels of detail (i.e., CSCIs, CSCs, CSUs, and procedures). In some manner, scores for those lower levels must also be considered in calculating the complete higher-level values. This section describes how, for a particular phase, this combination may be accomplished.

There are six levels to the software architecture hierarchy represented in the Software Quality Framework. These are, in ascending order:

- Procedure (The lowest-level of the architecture. This level may correspond, for example, to modules or to the elements in an Ada package)

- Package (An aggregation of procedures. No questions are asked at the package level.)

- Computer Software Unit (CSU)

- Computer Software Component (CSC)

- Computer Software Configuration Item (CSCI)

- System

Aggregation of scores from subordinate units in the software architecture hierarchy occurs in the metric-level scoring. Criterion and factor scores are computed for each level of the architecture as described in section 3.2.1. The following paragraphs describe, for each level, how metric scores may be aggregated across architecture levels.

3.2.2.1 Procedure-Level Scoring

The procedure is the bottom-most level in the software architecture hierarchy. Because the procedure is the smallest level of detail, procedure-level metric scoring does not involve metric elements collected from subordinate architecture levels. This means that procedure scores for metrics may be calculated for each procedure exactly as described in section 3.2.1. Scores are calculated individually for each procedure, and may be compared across procedures to locate potential trouble areas and identify non-compliant procedures.

Procedure scoring is possible only for Data Collection Forms *D* (Detailed Design) and *E* (Code and Unit Testing). These two DCFs are the only forms which collect procedure-level data.

An example of this scoring process for a particular procedure is calculation of the metric ID.2 (Machine Independence). The metric elements applicable to this metric at the Detailed Design phase and the procedure level are ID.2.3.d, ID.2.4.d, and ID.2.5.d. The calculation of the value for metric ID.2 would follow these steps:

1. For each *Y* answer in a metric element, score that element as 1.

2. For each *N* answer in a metric element, score that element as 0.

3. Calculate the metric score using the following equation, obtained from the formula lists in Appendix A:

   ID.2 = a*(ID.2.3.d) + b*(ID.2.4.d) + c*(ID.2.5.d)

   where:

   * indicates performing a multiplication

   (AA.n.n.a) represents the value of the particular metric element

   a, b, and c are multiplying coefficients which sum to the value 1. In this case, with no weighting performed, each coefficient will equal 1/3. Weighting can change the relative values, but the sum must always remain 1.

   For any metric element that is not applicable (scored as N/A), the coefficient is set to a value of 0.

3.2.2.2 Package-Level Scoring

The package is an aggregation of the procedure-level units in the software architecture hierarchy. Because there are no metric elements at the package level, a package-level metric score is the average of the scores of the subordinate procedures. To determine metric scores, the following rule is applied:

1. The metric is scored as the average of the metric scores for the constituent entities.

3.2.2.3 CSU-Level Scoring

The computer software unit is made up of the package-level entities described above. Metric elements at the CSU level as well as those that relate to the CSU's subordinate units are applicable to the CSU-level metric score. The metric score for the unit is computed from the CSU-level metric elements as described in section 3.2.1, and added to this is the average metric score of the subordinate units.

Computer software unit scoring is possible only for Data Collection Forms *D* (Detailed Design) and *E* (Code and Unit Testing). These two DCFs are the only forms which collect CSU-level data.

An example of this scoring process for a particular unit is calculation of the metric ST.1 (Interface Complexity). The metric elements applicable to this metric at the Detailed Design phase and the CSU level are ST.1.4.d and ST.1.5.d. Metric elements also exist for the procedure level (ST.1.1.d, ST.1.2.d, and ST.1.3.d). These procedure-level metric elements are used to compute the metric-level score for the procedures and in turn are aggregated into package-level metric scores. The calculation of the value for metric ST.1 for the CSU level would follow these steps:

1. For each *Y* answer in a metric element, score that element as 1.

2. For each *N* answer in a metric element, score that element as 0.

3. Calculate the metric score using the equation, obtained from the formula lists in Appendix A and add the average of the subordinate scores:

$$ST.1 = a*(ST.1.4.d) + b*(ST.1.5.d) + c*(AVG(ST.1)_{package})$$

where:

* indicates performing a multiplication

(AA.n.n.a) represents the value of the particular metric element

a, b, and c are multiplying coefficients which sum to the value 1. In this case, with no weighting performed, each coefficient will equal 1/3. Weighting can change the relative values, but the sum must always remain 1.

For any metric element that is not applicable (scored as N/A), the coefficient is set to a value of 0.

$AVG(ST.1)_{package}$ indicates the average ST.1 metric score for the subordinate packages.

3.2.2.4 CSC-Level Scoring

The computer software component level of the software architecture is composed of computer software units. The metric elements that are applicable at this level are, therefore, those that are directly related to the CSC and collected at that level, as well as those that relate to its subordinate CSUs.

An example of this scoring process for a particular unit is calculation of the metric AM.2 (Improper Input Data). The metric element applicable to this metric at the Detailed Design phase and the CSC level is AM.2.1.d. Metric elements also exist for the CSU level. The calculation of the value for metric AM.2 for the CSC level would follow these steps:

1. For each *Y* answer in a metric element, score that element as 1.

2. For each *N* answer in a metric element, score that element as 0.

3. Calculate the metric score using the equation, obtained from the formula lists in Appendix A and add the average of the subordinate scores:

$$AM.2 = a*(AM.2.1.d) + b*(AVG(AM.2)_{CSU})$$

where:

* indicates performing a multiplication

(AA.n.n.a) represents the value of the particular metric element

a and b are multiplying coefficients which sum to the value 1. In this case, with no weighting performed, each coefficient will equal 1/2. Weighting can change the relative values, but the sum must always remain 1.

For any metric element that is not applicable (scored as N/A), the coefficient is set to a value of 0.

$AVG(AM.2)_{CSU}$ indicates the average AM.2 metric score for the subordinate CSUs.

Of the nine Data Collection Forms in section 2, only DCFs *C* (Preliminary Design), *D* (Detailed Design), and *F* (CSC Integration and Test) are directly applicable to CSC-level metric elements. However, DCF *E* (Code and Unit Testing) collects data at the CSU level. Since CSC's are composed of CSUs, it is meaningful to score the CSC for that DCF as well.

DCF *C* (Preliminary Design) and *F* (CSC Integration and Testing) collect CSC-level data, but do not collect data on the subordinate CSUs. That means the CSC scoring for DCFs *C* and *F* is done exactly as described in section 3.2.1.

3.2.2.5 CSCI-Level Scoring

CSCIs are scored in the same fashion as CSCs. CSCI-level metric scores are computed from the scoring equation in Appendix A plus the average metric score of the subordinate CSCs.

3.2.2.6 System-Level Scoring

System level scores are calculated in the same fashion as CSCs. System-level metric scores are computed from the scoring equation in Appendix A plus the average metric score of the subordinate CSCIs.

3.2.3 Capability Scoring

Scores may be determined for system-level capabilities. To determine these scores, calculate the CSCI-level scores for the desired phase. For each capability, assess which CSCIs implement that capability, and the percentage of each CSCI that is related to the capability. For example, a user may wish to evaluate the Man-Machine Interface system-level capability. He might determine that two CSCIs implement that capability, and each contributes 50%. He would then average the two CSCI scores to obtain the capability scoring.

APPENDIX A -- METRIC SCORING FORMULAS BY ARCHITECTURE LEVEL AND PHASE

Introduction

This appendix presents the formulas used to calculate metric scores based on metric element results. These formulas calculate each metric for a particular phase and for a particular level of software architecture application. Metrics that are applied at multiple levels within a phase (e.g., metric elements that apply to both computer software components and computer software units) will not be reflected in these equations.

The following applies to each of the formulas listed:

* indicates a multiplication is to be performed

+ indicates an addition is to be performed

/ indicates a division is to be performed

(AA.n.n.a) represents the value for a particular metric element. Metric elements should be scored such that a *Y* answer gets a score of 1, a *N* answer gets a score of 0, and a numeric answer is used directly.

a, b, c, d, ... are multiplying coefficients which should sum to a value of 1 in each equation. Weighting can change the relative values of these coefficients, but they should always sum to 1. For any metric element that is N/A (not applicable), the coefficient should be 0. The default values of the coefficients are set to produce an average, with each coefficient equal to the others.

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
AC.1	a	System	a*(AC.1.1.a) + b*(AC.1.2.a) + c*(AC.1.4.a) + d*(AC.1.5.a) + e*(AC.1.6.a) + f*(AC.1.7.a)
	b	CSCI	a*(AC.1.4.b) + b*(AC.1.5.b) + c*(AC.1.6.b) + d*(AC.1.7.b)
	c	CSC	a*(AC.1.3.c) + b*(AC.1.4.c) + c*(AC.1.5.c) + d*(AC.1.6.c) + e*(AC.1.7.c)
	d	CSU	a*(AC.1.3.d) + b*(AC.1.4.d) + c*(AC.1.5.d) + d*(AC.1.6.d) + e*(AC.1.7.d)
	e	CSU	(AC.1.5.e)
	f	CSC	(AC.1.5.f)
	g	CSCI	(AC.1.5.g)
	h	System	(AC.1.5.h)
	i	System	(AC.1.5.i / 100)
AM.1	a	System	a*(AM.1.2.a / AM.1.1.a) + b*(AM.1.4.a / AM.1.3.a) + c*(AM.1.5.a) + d*(AM.1.7.a / AM.1.6.a)
	b	CSCI	a*(AM.1.2.b / AM.1.1.b) + b*(AM.1.4.b / AM.1.3.b) + c*(AM.1.5.b) + d*(AM.1.7.b / AM.1.6.b)
	c	CSC	a*(AM.1.4.c / AM.1.3.c) + b*(AM.1.5.c) + c*(AM.1.7.c / AM.1.6.c)
	d	CSU	(AM.1.5.d)
	e	CSU	(AM.1.5.e)
	f	CSC	a*(AM.1.4.f / AM.1.3.f) + b*(AM.1.5.f)
	g	CSCI	a*(AM.1.4.g / AM.1.3.g) + b*(AM.1.5.g)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	h	System	a*(AM.1.4.h / AM.1.3.h) + b*(AM.1.5.h)
AM.2	a	System	(AM.2.1.a)
	b	CSCI	(AM.2.1.b)
	c	CSC	a*(AM.2.1.c) + b*(AM.2.2.c) + c*(AM.2.3.c) + d*(AM.2.4.c) + e*(AM.2.5.c) + f*(AM.2.6.c)
	d	CSC	(AM.2.1.d)
	d	CSU	a*(AM.2.2.d) + b*(AM.2.3.d) + c*(AM.2.4.d) + d*(AM.2.5.d) + e*(AM.2.6.d)
	e	CSU	a*(AM.2.1.e) + b*(AM.2.3.e) + c*(AM.2.4.e) + d*(AM.2.5.e) + e*(AM.2.6.e)
	f	CSC	a*(AM.2.3.f) + b*(AM.2.4.f) + c*(AM.2.5.f) + c*(AM.2.6.f)
AM.3	a	System	(AM.3.1.a)
	b	CSCI	a*(AM.3.1.b) + b*(AM.3.2.b) + c*(AM.3.3.b) + d*(AM.3.4.b)
	c	CSCI	a*(AM.3.1.c) + b*(AM.3.2.c) + c*(AM.3.3.c) + d*(AM.3.4.c)
	d	CSU	a*(AM.3.1.d) + b*(AM.3.2.d) + c*(AM.3.3.d) + d*(AM.3.4.d)
	e	CSU	a*(AM.3.1.e) + b*(AM.3.2.e) + c*(AM.3.3.e) + d*(AM.3.4.e)
	f	CSC	a*(AM.3.1.f) + b*(AM.3.2.f) + c*(AM.3.3.f) + d*(AM.3.4.f)
	g	CSCI	a*(AM.3.1.g) + b*(AM.3.2.g) + c*(AM.3.3.g) + d*(AM.3.4.g)
	h	System	(AM.3.1.h)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	i	System	(AM.3.1.i / 100)
AM.4	a	System	(AM.4.1.a)
	b	CSCI	(AM.4.1.b)
	c	CSCI	(AM.4.1.c)
	d	CSC	(AM.4.1.d)
	e	CSU	(AM.4.1.e)
	f	CSC	(AM.4.1.f)
	g	CSCI	(AM.4.1.g)
	h	System	(AM.4.1.h)
	i	System	(AM.4.1.i / 100)
AM.5	a	System	(AM.5.1.a)
	b	CSCI	(AM.5.1.b)
	c	CSCI	(AM.5.1.c)
	d	CSC	(AM.5.1.d)
	e	CSC	(AM.5.1.e)
	f	CSC	(AM.5.1.f)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	g	CSCI	(AM.5.1.g)
	h	System	(AM.5.1.h)
	i	System	(AM.5.1.i / 100)
AM.6	a	System	(AM.6.1.a)
	b	CSCI	a*(AM.6.1.b) + b*(AM.6.2.b) + c*(AM.6.3.b) + d*(AM.6.4.b)
	c	CSC	a*(AM.6.2.c) + b*(AM.6.3.c) + c*(AM.6.4.c)
	c	CSCI	(AM.6.1.c)
	e	CSC	a*(AM.6.1.e) + b*(AM.6.2.e) + c*(AM.6.3.e) + d*(AM.6.4.e)
	f	CSC	a*(AM.6.1.f) + b*(AM.6.2.f) + c*(AM.6.3.f) + d*(AM.6.4.f)
	g	CSCI	a*(AM.6.1.g) + b*(AM.6.2.g) + c*(AM.6.3.g) + d*(AM.6.4.g)
	h	System	(AM.6.1.h)
	i	System	(AM.6.1.i / 100)
AM.7	a	System	a*(AM.7.1.a) + b*(AM.7.2.a) + c*(AM.7.3.a)
	b	CSCI	a*(AM.7.1.b) + b*(AM.7.2.b) + c*(AM.7.3.b)
	c	CSC	a*(AM.7.1.c) + b*(AM.7.2.c) + c*(AM.7.3.c)
	e	CSC	a*(AM.7.1.e) + b*(AM.7.2.e) + c*(AM.7.3.e)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	f	CSC	a*(AM.7.1.f) + b*(AM.7.2.f) + c*(AM.7.3.f)
	g	CSCI	a*(AM.7.1.g) + b*(AM.7.2.g) + c*(AM.7.3.g)
	h	System	a*(AM.7.1.h) + b*(AM.7.2.h) + c*(AM.7.3.h)
	i	System	a*(AM.7.1.i / 100) + b*(AM.7.3.i)
AP.1	a	System	(AP.1.1.a)
	b	CSCI	(AP.1.1.b)
	c	CSCI	(AP.1.1.c)
	d	CSU	(AP.1.1.d)
	e	CSU	(AP.1.1.e)
	f	CSC	(AP.1.1.f)
	g	CSCI	(AP.1.1.g)
AP.2	a	System	a*(AP.2.3.a) + b*(AP.2.4.a)
	b	CSCI	a*(AP.2.3.b) + b*(AP.2.4.b)
	d	Procedure	(AP.2.1.d) / (AP.2.1.d + AP.2.2.d)
	e	CSU	a*(AP.2.3.e) + b*(AP.2.4.e)
	e	Procedure	(AP.2.1.e) / (AP.2.1.e + AP.2.2.e)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
AP.3	a	System	(AP.3.1.a)
	b	CSCI	(AP.3.1.b)
	c	CSCI	(AP.3.1.c)
	d	CSC	(AP.3.1.d)
	e	Procedure	a*(AP.3.1.e) + b*(AP.3.3.e - AP.3.4.e) / (AP.3.3.e)
	f	CSC	(AP.3.1.f)
	g	CSCI	(AP.3.1.g)
	h	System	(AP.3.1.h)
AP.4	a	System	(AP.4.1.a)
	b	CSCI	(AP.4.1.b)
	c	CSCI	(AP.4.1.c)
	d	Procedure	(AP.4.1.d)
	e	Procedure	(AP.4.1.e)
	f	CSC	(AP.4.1.f)
	g	CSCI	(AP.4.1.g)
	h	System	(AP.4.1.h)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
AP.5	a	System	a*(AP.5.1.a) + b*(AP.5.3.a / AP.5.2.a)
	b	CSCI	a*(AP.5.1.b) + b*(AP.5.3.b / AP.5.2.b)
	c	CSCI	a*(AP.5.1.c) + b*(AP.5.4.c) + c*(AP.5.5.c)
	d	CSC	(AP.5.1.d)
	f	CSC	(AP.5.1.f)
	g	CSCI	(AP.5.1.g)
	h	System	(AP.5.1.h)
	i	System	(AP.5.1.i)
AT.1	a	System	a*(AT.1.2.a) + b*(AT.1.5.a)
	b	CSCI	a*(AT.1.2.b) + b*(AT.1.5.b)
	c	CSC	a*(1 - (AT.1.4.c / AT.1.3.c)) + b*(1 - AT.1.7.c / AT.1.6.c)
	d	CSC	a*(1 - (AT.1.4.d / AT.1.3.d)) + b*(1 - (AT.1.7.d / AT.1.6.d))
	d	CSU	(AT.1.1.d)
	e	CSU	(AT.1.1.e)
	f	CSC	a*(1 - (AT.1.4.f / AT.1.3.f)) + b*(1 - (AT.1.7.f / AT.1.6.f))
	g	CSCI	a*(1 - (AT.1.4.g / AT.1.3.g)) + b*(1 - (AT.1.7.g / AT.1.6.g))

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	h	System	a*(1 - (AT.1.4.h / AT.1.3.h)) + b*(1 - (AT.1.7.h / AT.1.6.h))
AT.2	a	System	(AT.2.3.a)
	b	CSCI	(AT.2.3.b)
	c	CSC	(1 - (AT.2.5.c / AT.2.4.c))
	d	CSC	(1 - (AT.2.5.d / AT.2.4.d))
	d	CSU	a*(AT.2.1.d) + b*(AT.2.2.d)
	e	Procedure	a*(AT.2.1.e) + b*(AT.2.2.e)
	f	CSC	(1 - (AT.2.5.f / AT.2.4.f))
	g	CSCI	(1 - (AT.2.5.g / AT.2.4.g))
	h	System	(1 - (AT.2.5.h / AT.2.4.h))
AT.3	a	System	a*(AT.3.1.a) + b*(AT.3.4.a)
	b	CSCI	a*(AT.3.1.b) + b*(AT.3.4.b)
	c	CSC	a*(1 - (AT.3.3.c / AT.3.2.c)) + b*(1 - (AT.3.6.c / AT.3.5.c))
	d	CSC	a*(1 - (AT.3.3.d / AT.3.2.d)) + b*(1 - (AT.3.6.d / AT.3.5.d))
	f	CSC	a*(1 - (AT.3.3.f / AT.3.2.f)) + b*(1 - (AT.3.6.f / AT.3.5.f))
	g	CSCI	a*(1 - (AT.3.3.g / AT.3.2.g)) + b*(1 - (AT.3.6.g / AT.3.5.g))

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	h	System	a*(1 - (AT.3.3.h / AT.3.2.h)) + b*(AT.3.6.h / AT.3.5.h)
AT.4	a	System	a*(AT.4.1.a) + b*(AT.4.2.a) + c*(AT.4.3.a)
	b	CSCI	a*(AT.4.1.b) + b*(AT.4.2.b) + c*(AT.4.3.b)
	c	CSCI	(AT.4.1.c)
	g	CSCI	(AT.4.1.g)
	h	System	(AT.4.1.h)
AU.1	d	CSCI	(1 - (AU.1.6.d / AU.1.7.d))
	d	CSU	(AU.1.5.d)
	d	Procedure	(1 - (AU.1.3.d / AU.1.2.d))
	e	CSU	(AU.1.5.e)
	e	Procedure	(AP.3.3.e - AU.1.3.e) / (AP.3.3.e)
	g	CSCI	a*(1 - (AU.1.6.g / AU.1.7.g))
AU.2	a	System	a*(AU.2.1.a) + b*(AU.2.2.a)
	b	CSCI	(AU.2.2.b)
	c	CSCI	(AU.2.2.c)
	g	CSCI	(AU.2.2.g)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	h	System	a*(AU.2.1.h) + b*(AU.2.2.h)
CL.1	a	System	a*(CL.1.1.a) + b*(CL.1.2.a) + c*(1 - CL.1.3.a / AP.5.2.a) + d*(1 - CL.1.4.a / AP.5.2.a) + e*(CL.1.5.a) + f*(CL.1.6.a) + g*(CL.1.7.a) + h*(CL.1.8.a) + i*(CL.1.9.a) + j*(CL.1.10.a) + k*(CL.1.11.a) + l*(CL.1.12.a) + m*(1 - CL.1.13.a / CL.1.16.a) + n*(CL.1.14.a) + o*(CL.1.15.a) + p*(CL.1.17.a) + q*(CL.1.18.a)
	b	CSCI	a*(CL.1.1.b) + b*(CL.1.2.b) + c*(1 - CL.1.3.b / AP.5.2.b) + d*(1 - CL.1.4.b / AP.5.2.b) + e*(CL.1.5.b) + f*(CL.1.6.b) + g*(CL.1.7.b) + h*(CL.1.8.b) + i*(CL.1.9.b) + j*(CL.1.10.b) + k*(CL.1.11.b) + l*(CL.1.12.b) + m*(1 - (CL.1.13.b / CL.1.16.b)) + n*(CL.1.14.b) + o*(CL.1.15.b) + p*(1 / CL.1.16.b) + q*(CL.1.17.b) + r*(CL.1.18.b)
	c	CSCI	a*(CL.1.6.c) + b*(CL.1.7.c) + c*(CL.1.8.c) + d*(CL.1.10.c) + e*(CL.1.11.c) + f*(CL.12.c) + g*(CL.1.17.c) + h*(CL.1.18.c) + i*(CL.1.19.c)
	c	CSC	a*(CL.1.3.c) + b*(CL.1.4.c)
	d	CSU	a*(CL.1.3.d) + b*(CL.1.4.d)
	e	CSU	a*(CL.1.3.e) + b*(CL.1.4.e)
	g	CSCI	a*(CL.1.1.g) + b*(CL.1.2.g) + c*(CL.1.5.g) + d*(CL.1.6.g) + e*(CL.1.7.g) + f*(CL.1.8.g) + g*(CL.1.9.g) + h*(CL.1.10.g) + i*(CL.1.12.g) + h*(CL.1.17.g) + i*(CL.1.18.g)
	h	System	a*(CL.1.1.h) + b*(CL.1.2.h) + c*(CL.1.5.h) + d*(CL.1.6.h) + e*(CL.1.7.h) + f*(CL.1.8.h) + g*(CL.1.9.h) + h*(CL.1.10.h) + i*(CL.1.11.h) + j*(CL.1.12.h) + k*(CL.1.17.h) + l*(CL.1.18.h)
	i	System	a*(CL.1.17.i) + b*(CL.1.18.i)
CL.2	a	System	a*(CL.2.1.a) + b*(CL.2.2.a) + c*(1 - CL.2.3.a / AP.5.2.a) + d*(CL.2.4.a) + e*(1/ CL.2.5.a) + f*(CL.2.6.a) + g*(1 / CL.2.7.a) + h*(CL.2.9.a / CL.2.8.a) + i*(CL.2.11.a / CL.2.10.a)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	b	CSCI	a*(CL.2.1.b) + b*(CL.2.2.b) + c*(1 - CL.2.3.b / AP.5.2.b) + d*(CL.2.4.b) + e*(1 / CL.2.5.b) + f*(CL.2.6.b) + g*(1 / CL.2.7.b) + h*(CL.2.9.b / CL.2.8.b) + i*(CL.2.11.b / CL.2.10.b)
	c	CSCI	a*(1 / CL.2.5.c) + b*(1 / CL.2.7.c) + c*(CL.2.9.c / CL.2.8.c) + d*(CL.2.11.c / CL.2.10.c)
	c	CSC	a*(CL.2.2.c) + b*(CL.2.3.c)
	d	CSU	a*(CL.2.2.d) + b*(CL.2.3.d)
	e	CSU	a*(CL.2.2.e) + b*(CL.2.3.e)
	f	CSC	(CL.2.2.f)
	f	CSU	(CL.2.3.f)
	g	CSCI	a*(CL.2.2.g) + b*(1 / CL.2.3.g) + c*(1 / (1 + CL.2.5.g)) + d*(1 / (1 + CL.2.7.g)) + e*(CL.2.9.g / CL.2.8.g) + f*(CL.2.11.g / CL.2.10.g)
	h	System	a*(CL.2.2.h) + b*(1 / CL.2.3.h) + c*(CL.2.4.h) + d*(1 / CL.2.5.h) + e*(CL.2.6.h) + f*(1 / (1 + CL.2.7.h)) + g*(CL.2.9.h / CL.2.8.h) + h*(CL.2.11.h / CL.2.10.h)
CL.3	a	System	(CL.3.1.a)
	b	CSCI	(CL.3.1.b)
CP.1	a	System	a*(CP.1.1.a / AP.5.2.a) + b*(CP.1.3.a / CP.1.2.a) + c*(1 - (CP.1.4.a / CP.1.2.a)) + d*(CP.1.5.a / CP.1.2.a) + e*(CP.1.6.a / CP.1.2.a) + f*(CP.1.7.a/ AP.5.2.a) + g*(CP.1.8.a / AP.5.2.a) + h*(CP.1.9.a / AP.5.2.a) + i*(CP.1.10.a / AP.5.2.a) + j*(CP.1.13.a / CP.1.12.a)
	b	CSCI	a*(CP.1.1.b / AP.5.2.b) + b*(CP.1.3.b / CP.1.2.b) + c*(1 - (CP.1.4.b / CP.1.2.b)) + d*(CP.1.5.b / CP.1.2.b) + e*(CP.1.6.b / CP.1.2.b) + f*(CP.1.7.b) + g*(CP.1.8.b / AP.5.2.b) + h*(CP.1.9.b / CP.1.8.b) + i*(CP.1.10.b / AP.5.2.b) + j*(CP.1.13.b / CP.1.12.b)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSCI	a*(CP.1.7.c) + b*(CP.1.13.c / CP.1.12.c)
	c	CSC	a*(CP.1.1.c) + b*(CP.1.3.c / CP.1.2.c) + c*(1 - (CP.1.4.c / CP.1.2.c)) + d*(CP.1.5.c / CP.1.2.c) + e*(CP.1.6.c / CP.1.2.c) + f*(CP.1.10.c)
	d	CSCI	(CP.1.13.d / CP.1.12.d)
	d	CSC	(CP.1.7.d)
	d	CSU	a*(CP.1.1.d) + b*(CP.1.10.d)
	d	Procedure	a*(CP.1.3.d / CP.1.2.d) + b*(1 - (CP.1.4.d / CP.1.2.d)) + c*(CP.1.5.d / CP.1.2.d)+ d*(CP.1.6.d / CP.1.2.d) + e*(CP.1.11.d)
	e	CSCI	(CP.1.13.e / CP.1.12.e)
	e	CSU	a*(CP.1.1.e) + b*(CP.1.10.e)
	e	Procedure	a*(CP.1.3.e / CP.1.2.e) + b*(CP.1.4.e / CP.1.2.e) + c*(CP.1.5.e / CP.1.2.e) + d*(CP.1.6.e / CP.1.5.e) + f*(CP.1.11.e)
	f	CSCI	(CP.1.13.f / CP.1.12.f)
	g	CSCI	(CP.1 13.g / CP.1.12.g)
	h	System	(CP.1.13.h / CP.1.12.h)
	i	System	(CP.1.13.i / CP.1.12.i)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
CS.1	a	System	a*(CS.1.1.a) + b*(CS.1.2.a) + c*(CS.1.4.a) + d*(CS.1.5.a) + e*(CS.1.7.a) + f*(CS.1.8.a) + g*(CS.1.10.a) + h*(CS.1.11.a) + i*(CS.1.12.a)
	b	CSCI	a*(CS.1.1.b) + b*(CS.1.2.b) + c*(CS.1.4.b) + d*(CS.1.5.b) + e*(CS.1.7.b) + f*(CS.1.8.b) + g*(CS.1.10.b) + h*(CS.1.11.b) + i*(CS.1.12.b / AP.5.2.b)
	c	CSCI	a*(CS.1.2.c) + b*(CS.1.5.c) + c*(CS.1.8.c) +d*(CS.1.11.c) + e*(CS.1.12.c / AP.5.6.c)
	d	CSCI	(CS.1.2.d)
	d	CSC	(CS.1.12.d / AP.5.7.d)
	d	CSU	a*(CS.1.5.d) + b*(CS.1.8.d) + c*(CS.1.11.d)
	e	CSC	(CS.1.12.e / AP.5.7.e)
	e	CSU	a*(CS.1.4.e) + b*(CS.1.5.e) + c*(CS.1.8.e) + d*(CS.1.11.e)
CS.2	a	System	a*(CS.2.1.a) + b*(CS.2.2.a) + c*(CS.2.4.a) + d*(CS.2.5.a) + e*(CS.2.7.a) + f*(CS.2.8.a) + g*(CS.2.11.a) + h*(CS.2.13.a) + i*(CS.2.14.a)
	b	CSCI	a*(CS.2.1.b) + b*(CS.2.2.b) + c*(CS.2.4.b) + d*(CS.2.5.b) + e*(CS.2.7.b) + f*(CS.2.8.b) + g*(CS.2.11.b) + h*(CS.2.13.b) + i*(CS.2.14.b / CP.1.3.b)
	c	CSCI	a*(CS.2.2.c) + b*(CS.2.11.c) + c*(CS.2.13.c)
	c	CSC	a*(CS.2.5.c) + b*(CS.2.8.c) + c*(CS.2.14.c)
	d	CSU	a*(CS.2.2.d) + b*(CS.2.5.d) + c*(CS.2.8.d) + d*(CS.2.14.d)
	e	CSU	a*(CS.2.5.e) + b*(CS.2.8.e) + c*(CS.2.14.e)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	f	CSC	(CS.2.14.f)
	g	CSCI	a*(CS.2.11.g) + b*(CS.2.13.g) + c*(CS.2.14.g)
	h	System	a*(CS.2.11.h) + b*(CS.2.13.h) + c*(CS.2.14.h)
	i	System	(CS.2.11.i)
DI.1	a	System	a*(DI.1.1.a) + b*(DI.1.2.a) + c*(DI.1.3.a) + d*(DI.1.4.a) + e*(DI.1.5.a) + f*(DI.1.6.a) + g*(DI.1.8.a) + h*(DI.1.9.a) + i*(DI.1.10.a)
	b	CSCI	a*(DI.1.1.b) + b*(DI.1.2.b) + c*(DI.1.3.b) + d*(DI.1.4.b) + e*(DI.1.6.b) + f*(DI.1.8.b) + g*(DI.1.9.b) + h*(DI.1.10.b)
	c	CSCI	a*(DI.1.1.c) + b*(DI.1.3.c) + c*(DI.1.4.c) + d*(DI.1.5.c) + e*(DI.1.6.c) +f*(DI.1.8.c) + g*(DI.1.9.c) + h*(DI.1.10.c)
	c	CSC	(DI.1.2.c)
	d	CSCI	a*(DI.1.1.d) + b*(DI.1.2.d) + c*(DI.1.3.d) + d*(DI.1.4.d) + e*(DI.1.5.d) + f*(DI.1.6.d) + g*(DI.1.8.d) + h*(DI.1.9.d) + i*(DI.1.10.d)
	e	CSCI	(DI.1.5.e)
	g	CSCI	a*(DI.1.3.g) + b*(DI.1.4.g) + c*(DI.1.5.g) + d*(DI.1.6.g) + e*(DI.1.8.g) + f*(DI.1.9.g) + g*(DI.1.10.g)
	h	System	a*(DI.1.3.h) + b*(DI.1.4.h) + c*(DI.1.5.h) + d*(DI.1.6.h) + e*(DI.1.8.h) + f*(DI.1.9.h) + g*(DI.1.10.h)
	i	System	a*(DI.1.5.i) + b*(DI.1.10.i / 100)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
DO.1	a	System	(DO.1.1.a)
	b	CSCI	(DO.1.1.b)
	c	CSCI	(DO.1.1.c)
	d	CSCI	(DO.1.1.d)
	e	CSCI	(DO.1.1.e)
	f	CSCI	(DO.1.1.f)
	g	CSCI	(DO.1.1.g)
	h	System	(DO.1.1.h)
DO.2	a	System	a*(DO.2.1.a) + b*(DO.2.2.a) + c*(DO.2.3.a) + d*(DO.2.4.a) + e*(DO.2.5.a) + f*(DO.2.6.a)
	b	CSCI	a*(DO.2.1.b) + b*(DO.2.2.b) + c*(DO.2.3.b) + d*(DO.2.4.b) + e*(DO.2.5.b) + f*(DO.2.6.b)
	c	CSCI	a*(DO.2.1.c) + b*(DO.2.2.c) + c*(DO.2.3.c) + d*(DO.2.4.c) + e*(DO.2.5.c) + f*(DO.2.6.c) + g*(DO.2.7.c)
	d	CSCI	a*(DO.2.1.d) + b*(DO.2.2.d) + c*(DO.2.3.d) + d*(DO.2.4.d) + e*(DO.2.5.d) + f*(DO.2.7.d)
	d	CSC	(DO.2.6.d)
	e	CSCI	a*(DO.2.1.e) + b*(DO.2.2.e) + c*(DO.2.3.e) + d*(DO.2.4.e)
	e	CSU	(DO.2.6.e)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	f	CSCI	a*(DO.2.1.f) + b*(DO.2.3.f) + c*(DO.2.4.f)
	g	CSCI	a*(DO.2.1.g) + b*(DO.2.3.g) + c*(DO.2.4.g) + d*(DO.2.8.g)
	h	System	a*(DO.2.1.h) + b*(DO.2.3.h) + c*(DO.2.4.h) + d*(DO.2.8.h)
	i	System	a*(DO.2.1.i) + b*(DO.2.2.i) + c*(DO.2.3.i) + d*(DO.2.5.i) + e*(DO.2.8.i)
EC.1	a	System	(EC.1.1.a)
	b	CSCI	(EC.1.1.b)
	c	CSC	(EC.1.1.c)
	f	CSC	(EC.1.1.f)
	g	CSCI	(EC.1.1.g)
	h	System	(EC.1.1.h)
EP.1	a	System	a*(EP.1.1.a) + b*(EP.1.3.a) + c*(EP.1.9.a)
	b	CSCI	a*(EP.1.1.b) + b*(EP.1.3.b) + c*(EP.1.9.b)
	c	CSCI	(EP.1.9.c)
	c	CSC	(EP.1.1.c)
	d	CSU	(EP.1.1.d)
	d	Procedure	a*(1 - (EP.1.6.d / EP.1.5.d)) + b*(1 - (EP.1.8.d / EP.1.7.d)) + c*(1 - (EP.1.11.d / EP.1.10.d))

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	CSCI	(1 / EP.1.9.e)
	e	CSU	a*(EP.1.1.e) + b*(EP.1.3.e)
	e	Procedure	a*(1 - (EP.1.6.e / EP.1.5.e)) + b*(1 - (EP.1.8.e / EP.1.7.e)) + c*(1 - (EP.1.11.e / EP.1.10.e))
	f	CSC	(EP.1.1.f)
	g	CSCI	a*(EP.1.1.g) + b*(1 / EP.1.9.g)
	h	System	a*(EP.1.1.h) + b*(EP.1.9.h)
EP.2	a	System	a*(EP.2.1.a) + b*(EP.2.2.a) + c*(EP.2.9.a)
	b	CSCI	a*(EP.2.1.b) + b*(EP.2.2.b) + c*(EP.2.9.b)
	c	CSCI	a*(EP.2.1.c) + b*(EP.2.2.c) + c*(EP.2.8.c)
	d	CSCI	a*(EP.2.1.d) + b*(EP.2.8.d)
	d	Procedure	a*(1 - (EP.2.4.d / EP.2.3.d)) + b*(1 - (EP.2.5.d / EP.2.3.d)) + c*(1 - (EP.2.7.d / CP.1.2.d))
	e	Procedure	a*(1 - (EP.2.4.e / EP.2.3.e)) + b*(1 - (EP.2.5.e / EP.2.3.e)) + c*(1 - (EP.2.7.e / EP.2.6.e))
	f	CSCI	(EP.2.1.f)
	g	CSCI	(EP.2.1.g)
	h	System	(EP.2.1.h)
ES.1	a	System	a*(ES.1.1.a) + b*(ES.1.2.a) + c*(ES.1.3.a) + d*(ES.1.4.a) + e*(ES.1.11.a) + f*(ES.1.12.a)
	b	CSCI	a*(ES.1.1.b) + b*(ES.1.2.b) + c*(ES.1.3.b) + d*(ES.1.4.b) + e*(ES.1.11.b) + f*(ES.1.12.b)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSCI	a*(ES.1.2.c) + b*(ES.1.3.c) + c*(ES.1.11.c) + d*(ES.1.12.c)
	d	CSCI	a*(1 - (ES.1.6.d / ES.1.5.d)) + b*(ES.1.7.d) + c*(ES.1.11.d)
	d	Procedure	(ES.1.8.d)
	e	CSCI	a*(1 - (ES.1.6.e / ES.1.5.e)) + b*(ES.1.7.e) + d*(ES.1.11.e)
	e	CSU	(ES.1.9.e)
	e	Procedure	(ES.1.8.e)
	g	CSCI	a*(ES.1.1.g) + b*(ES.1.4.g) + c*(ES.1.11.g)
	h	System	a*(ES.1.1.h) + b*(ES.1.4.h) + c*(ES.1.11.h)
FO.1	a	System	a*(1 - (FO.1.3.a / AP.5.2.a)) + b*(FO.1.4.a / FO.1.3.a) + c*(1 - (FO.1.5.a/ FO.1.3.a)) + d*(1 - (FO.1.6.a/FO.1.3.a))
	b	CSCI	a*(1 - (FO.1.3.b / AP.5.2.b)) + b*(FO.1.4.b / FO.1.3.b) + c*(1 - (FO.1.5.b / FO.1.3.b)) + d*(1 - (FO.1.6.b / FO.1.3.b))
	h	System	(FO.1.4.h / FO.1.3.h)
FS.1	d	CSU	(FS.1.1.d)
	e	CSU	a*(FS.1.1.e) + b*(FS.1.2.e)
FS.2	a	System	a*(FS.2.1.a) + b*(FS.2.3.a/AP.5.2.a) + c*(FS.2.4.a) + d*(FS.2.5.a) +e*(FS.2.6.a)
	b	CSCI	a*(FS.2.1.b) + b*(FS.2.3.b / AP.5.2.b) + c*(FS.2.4.b) + d*(FS.2.5.b) + e*(FS.2.6.b)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSC	a*(FS.2.4.c) + b*(FS.2.5.c) + c*(FS.2.6.c) + d*(FS.2.7.c / AP.5.2.c)
	c	CSCI	(FS.2.1.c)
	d	CSC	(FS.2.1.d)
	i	System	(FS.2.7.i / AP.5.2.i)
FS.3	a	System	a*(FS.3.1.a) + b*(FS.3.2.a)
	b	CSCI	a*(FS.3.1.b) + b*(FS.3.2.b)
	c	CSCI	a*(FS.3.1.c) + b*(FS.3.2.c)
	g	CSCI	a*(FS.3.1.g) + b*(FS.3.2.g)
	h	System	a*(FS.3.1.h) + b*(FS.3.2.h)
	i	System	a*(FS.3.1.i) + b*(FS.3.2.i)
GE.1	c	CSCI	(GE.1.1.c)
	d	CSCI	(GE.1.1.d / SUM (AP.5.7.d))
	e	CSCI	(GE.1.1.e / SUM(AP.5.7.e))
GE.2	a	System	a*(GE.2.1.a) + b*(GE.2.3.a) + c*(GE.2.4.a) + d*(GE.2.5.a)
	b	CSCI	a*(GE.2.1.b) + b*(GE.2.3.b) + c*(GE.2.4.b) + d*(GE.2.5.b)
	c	CSCI	a*(GE.2.1.c) + b*(GE.2.3.c) + c*(GE.2.4.c) + d*(GE.2.5.c)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	d	CSU	a*(GE.2.1.d) + b*(GE.2.3.d) + c*(GE.2.4.d) + d*(GE.2.5.d)
	e	CSU	a*(GE.2.1.e) + b*(GE.2.3.e) + c*(GE.2.4.e) + d*(GE.2.5.e)
	f	CSC	a*(GE.2.3.f) + b*(GE.2.4.f) + c*(GE.2.5.f)
	g	CSCI	a*(GE.2.3.g) + b*(GE.2.4.g) + c*(GE.2.5.g)
	h	System	a*(GE.2.3.h) + b*(GE.2.4.h) + c*(GE.2.5.h)
ID.1	a	System	a*(ID.1.1.a) + b*(ID.1.2.a)
	b	CSCI	a*(ID.1.1.b) + b*(ID.1.2.b)
	c	CSC	(ID.1.5.c)
	d	Procedure	a*(1 - (ID.1.3.d / AU.1.2.d)) + b*(ID.1.5.d)
	e	Procedure	a*(ID.1.1.e) + b*(1 - (ID.1.3.e / AP.3.3.e)) + c*(ID.1.5.e)
ID.2	a	System	(ID.2.1.a)
	b	CSCI	(ID.2.1.b)
	c	CSCI	(ID.2.1.c)
	d	Procedure	a*(ID.2.3.d) + b*(ID.2.4.d) + c*(ID.2.5.d)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	CSU	(ID.2.4.e)
	e	Procedure	a*(ID.2.3.e) + b*(ID.2.5.e)
MO.1	a	System	a*(MO.1.1.a) + b*(MO.1.2.a)
	b	CSCI	a*(MO.1.1.b) + b*(MO.1.2.b)
	c	CSC	a*(MO.1.2.c) + b*(MO.1.3.c)
	d	CSU	a*(MO.1.2.d) + b*(MO.1.3.d) + c*(MO.1.4.d) + d*(MO.1.10.d)
	d	Procedure	a*(1 - (MO.1.6.d / MO.1.5.d)) + b*(MO.1.7.d) + c*(MO.1.8.d) + d*(MO.1.9.d)
	e	CSU	a*(MO.1.2.e) + b*(MO.1.3.e) + c*(MO.1.10.e)
	e	Procedure	a*(MO.1.4.e) + b*(1 - (MO.1.6.e / MO.1.5.e)) + c*(MO.1.7.e) + d*(MO.1.8.e) + e*(MO.1.9.e)
	f	CSC	(MO.1.2.f)
	g	CSCI	(MO.1.2.g)
	h	System	(MO.1.2.h)
MO.2	a	System	a*(MO.2.1.a) + b*(MO.2.4.a)
	b	CSCI	a*(MO.2.1.b) + b*(1 - ((MO.2.3.b1 + MO.2.3.b2 + MO.2.3.b3) / (MO.2.2.b))) + c*(1 - ((MO.2.3.b4 + MO.2.3.b5) / (2*MO.2.2.b))) + d*(MO.2.3.b6 / MO.2.2.b) + e*(MO.2.4.b) + f*(MO.2.5.b)
	c	CSCI	a*(1 - ((MO.2.3.c1 + MO.2.3.c2 + MO.2.3.c3) / (MO.2.2.c))) + b*(1 - ((MO.2.3.c4 + MO.2.3.c5) / (2*MO.2.2.c))) + c*(MO.2.3.c6 / MO.2.2.c)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSC	(MO.2.5.c)
	d	CSCI	a*(1 - ((MO.2.3.d1 + MO.2.3.d2 + MO.2.3.d3) / (MO.2.2.d))) + b*(1 - ((MO.2.3.d4 + MO.2.3.d5) / (2*MO.2.2.d))) + c*(MO.2.3.d6 / MO.2.2.d)
	d	CSU	(MO.2.5.d)
	e	CSCI	a*(1 - ((MO.2.3.e1 + MO.2.3.e2 + MO.2.3.e3) / (MO.2.2.e))) + b*(1 - ((MO.2.3.e4 + MO.2.3.e5) / (2*MO.2.2.e))) + c*(MO.2.3.e6 / MO.2.2.e)
	e	CSU	(MO.2.5.e)
OP.1	a	System	a*(OP.1.1.a) + b*(OP.1.2.a) + c*(OP.1.3.a) + d*(OP.1.4.a) + e*(1 / (OP.1.5.a + 1)) + f*(OP.1.6.a) + g*(OP.1.7.a) + h*(OP.1.8.a) + i*(OP.1.10.a) + j*(OP.1.11.a) + k*(OP.1.12.a) + l*(OP.1.13.a) + m*(OP.1.14.a) + n*(OP.1.15.a)
	b	CSCI	a*(OP.1.1.b) + b*(OP.1.2.b) + c*(OP.1.3.b) + d*(OP.1.4.b) + e*(1/(1 + OP.1.5.b)) + f*(OP.1.6.b) + g*(OP.1.7.b) + h*(OP.1.8.b) + i*(OP.1.10.b) +j*(OP.1.11.b) + k*(OP.1.12.b) + l*(OP.1.13.b) + m*(OP.1.14.b) + n*(OP.1.15.b)
	c	CSCI	a*(OP.1.1.c) + b*(OP.1.2.c) + c*(OP.1.3.c) + d*(OP.1.4.c) + e*(OP.1.6.c) + f*(OP.1.7.c) + h*(OP.1.10.c) + i*(OP.1.11.c) + j*(OP.1.12.c) + k*(OP.1.13.c) + l*(OP.1.14.c) + n*(OP.1.15.c)
	d	CSCI	a*(1 - (OP.1.8.d / OP.1.9.d)) + b*(OP.1.17.d) + c*(OP.1.18.d)
	e	CSCI	(1 - (OP.1.8.e / OP.1.9.e))
	g	CSCI	a*(OP.1.1.g) + b*(OP.1.2.g) + c*(OP.1.3.g) + d*(OP.1.4.g) + e*(OP.1.6.g) + f*(OP.1.7.g) + g*(OP.1.8.g) + h*(OP.1.10.g) + i*(OP.1.11.g) + j*(OP.1.12.g) + k*(OP.1.13.g) + l*(OP.1.14.g) + m*(OP.1.15.g)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	h	System	a*(OP.1.1.h) + b*(OP.1.2.h) + c*(OP.1.3.h) + d*(OP.1.6.h) + e*(OP.1.7.h) + f*(OP.1.8.h) + g*(OP.1.10.h) + h*(OP.1.11.h) + i*(OP.1.12.h) + j*(OP.1.13.h) + k*(OP.1.14.h) + l*(OP.1.15.h)
	i	System	a*(OP.1.1.i) + b*(OP.1.2.i) + c*(OP.1.3.i) + d*(1 / OP.1.5.i) + e*(OP.1.6.i) + f *(OP.1.7.i) + g*(OP.1.8.i) + h*(OP.1.10.i) + i*(OP.1.11.i) + j*(OP.1.12.i) + k*(OP.1.13.i) + l*(OP.1.14.i) + m*(OP.1.15.i)
OP.2	a	System	a*(OP.2.2.a / OP.2.1.a) + b*(1 / OP.2.3.a) + c*(OP.2.4.a / OP.2.1.a) + d*(OP.2.6.a) + e*(OP.2.7.a) + f*(OP.2.8.a) + g*(OP.2.9.a)
	b	CSCI	a*(OP.2.2.b / OP.2.1.b) + b*(1 / OP.2.3.b) + c*(OP.2.4.b / OP.2.1.b) + d*(OP.2.6.b) + e*(OP.2.7.b) + f*(OP.2.8.b)
	c	CSCI	a*(OP.2.6.c) + b*(OP.2.7.c) + c*(OP.2.8.c)
	g	CSCI	a*(OP.2.2.g / OP.2.1.g) + b*(1 - (OP.2.3.g / OP.2.1.g)) + c*(OP.2.4.g / OP.2.1.g) + d*(OP.2.6.g) + e*(OP.2.7.g) + f*(OP.2.8.g)
	h	System	a*(OP.2.2.h / OP.2.1.h) + b*(1 - (OP.2.3.h / OP.2.1.h)) + c*(OP.2.4.h / OP.2.1.h) + d*(OP.2.6.h) + e*(OP.2.7.h) + f*(OP.2.8.h)
	i	System	a*(OP.2.2.i / OP.2.1.i) + b*(1 - (OP.2.3.i / OP.2.1.i)) + c*(OP.2.4.i / OP.2.1.i) + d*(OP.2.6.i) + e*(OP.2.7.i) + f*(OP.2.8.i)
OP.3	a	System	a*(OP.3.1.a) + b*(OP.3.2.a) + c*(OP.3.3.a) + d*(1 / OP.3.4.a) + e*(OP.3.5.a) + f*(OP.3.6.a) + g*(OP.3.7.a) + h*(OP.3.8.a)
	b	CSCI	a*(OP.3.1.b) + b*(OP.3.2.b) + c*(OP.3.3.b) + d*(1 / OP.3.4.b) + e*(OP.3.5.b) + f*(OP.3.6.b) + g*(OP.3.7.b) + h*(OP.3.8.b)
	c	CSCI	a*(OP.3.1.c) + b*(OP.3.2.c) + c*(OP.3.3.c) + d*(1 / OP.3.4.c) + e*(OP.3.5.c) + f*(OP.3.6.c)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	g	CSCI	a*(OP.3.1.g) + b*(OP.3.2.g) + c*(OP.3.3.g) + d*(1 / OP.3.4.g) + e*(OP.3.5.g) + f*(OP.3.6.g) + g*(OP.3.7.g) + h*(OP.3.8.g)
	h	System	a*(OP.3.1.h) + b*(OP.3.2.h) + c*(OP.3.3.h) + d*(1 / OP.3.4.h) + e*(OP.3.5.h) + f*(OP.3.6.h) + g*(OP.3.7.h) + h*(OP.3.8.h)
	i	System	a*(OP.3.1.i) + b*(OP.3.2.i) + c*(OP.3.3.i) + d*(1 / OP.3.4.i) + e*(OP.3.5.i) + f*(OP.3.6.i) + g*(OP.3.7.i) + h*(OP.3.8.i)
RE.1	a	System	a*(RE.1.1.a) + b*(RE.1.2.a) + c*(RE.1.3.a) + d*(RE.1.4.a)
	b	CSCI	a*(RE.1.1.b) + b*(RE.1.2.b) + c*(RE.1.3.b) + d*(RE.1.4.b)
	c	CSCI	a*(RE.1.1.c) + b*(RE.1.2.c) + c*(RE.1.3.c) + d*(RE.1.4.c)
	g	CSCI	a*(RE.1.1.g) + b*(RE.1.2.g) + c*(RE.1.3.g) + d*(RE.1.4.g)
	h	System	a*(RE.1.1.h) + b*(RE.1.2.h) + c*(RE.1.3.h) + d*(RE.1.4.h)
	i	System	a*(RE.1.1.i) + b*(RE.1.2.i) + c*(RE.1.3.i)
SD.1	e	Procedure	a*(SD.1.2.e / AP.3.3.e) + b*(SD.1.3.e / AP.3.3.e)
SD.2	a	System	a*(SD.2.1.a) + b*(SD.2.2.a)
	b	CSCI	a*(SD.2.1.b) + b*(SD.2.2.b)
	e	CSU	a*(SD.2.1.e) + b*(SD.2.2.e) + c*(SD.2.3.e) + d*(SD.2.4.e) + e*(SD.2.5.e) + f*(SD.2.6.e) + g*(SD.2.7.e) + h*(SD.2.8.e)
SD.3	a	System	(SD.3.7.a)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	b	CSCI	(SD.3.7.b)
	c	CSC	(SD.3.7.c)
	e	CSU	a*(SD.3.2.e) + b*(SD.3.3.e) + c*(SD.3.7.e)
	e	Procedure	a*(SD.3.1.e) + b*(1 - (SD.3.4.e / AP.3.3.e)) + c*(SD.3.5.e / AP.3.3.e) + d*(SD.3.8.e)
SI.1	a	System	a*(SI.1.1.a) + b*(SI.1.9.a) + c*(SI.1.10.a)
	b	CSCI	a*(SI.1.1.b) + b*(SI.1.9.b) + c*(SI.1.10.b)
	c	CSCI	a*(SI.1.1.c) + b*(1 - (SI.1.7.c / SUM(CP.1.2.c)) + c*(SI.1.9.c) + d*(1 / SI.1.8.c) + e*(SI.1.10.c)
	c	CSC	(SI.1.11.c)
	d	CSCI	a*(SI.1.1.d) + b*(SI.1.8.d / SI.1.7.d2)
	d	CSU	a*(SI.1.3.d) + b*(SI.1.4.d) + c*(SI.1.11.d)
	d	Procedure	a*(SI.1.2.d) + b*(1 / SI.1.5.d) + c*(1 / SI.1.6.d) + d*(1 - (SI.1.7.d1 / CP.1.2.d))
	e	CSCI	(SI.1.8.e / SI.1.7.e)
	e	CSU	a*(SI.1.2.e) + b*(SI.1.3.e) + c*(SI.1.4.e) + d*(SI.1.11.e)
	e	Procedure	a*(1 / SI.1.5.e) + b*(1 / SI.1.6.e)
SI.2	a	System	(SI.2.1.a)
	b	CSCI	(SI.2.1.b)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSCI	(SI.2.1.c)
	d	Procedure	(SI.2.1.d)
	g	CSCI	(SI.2.1.g)
	h	System	(SI.2.1.h)
SI.3	d	Procedure	1 / (1 + SI.3.1.d + SI.3.2.d)
	e	Procedure	1 / (1 + SI.3.1.e + SI.3.2.e)
SI.4	d	CSCI	(SI.4.18.d)
	d	CSU	a*(SI.4.8.d) + b*(SI.4.16.d)
	d	Procedure	a*(SI.4.1.d) + b*(1 - (SI.4.3.d / AU.1.2.d)) + c*(1 - (SI 4.5.d / SI.4.6.d)) + d*(1 - (SI.4.7.d / SI.4.6.d)) + e*(1 / SI.4.10.d) + f*(1 - (SI.4.11.d / AU.1.2.d)) + g*(1 - ((SI.4.12.d + SI.4.13.d)/ AU.1.2.d)) + h*(SI.4.15.d / SI.4.14.d)
	e	CSCI	(SI.4.18.e)
	e	CSU	a*(SI.4.8.e) + b*(SI.4.16.e) + c*(SI.4.17.e)
	e	Procedure	a*(SI.4.1.e) + b*(1 / (1 + SI.4.3.e)) + c*(1 - (SI.4.5.e / SI.4.6.e)) + e*(1 - (SI.4.7.e / SI.4.6.e)) + g*(1 - (SI.4.9.e / AP.3.3.e)) + h*(1 / SI.4.10.e) + i*(1 - (SI.4.11.e / AP.3.3.e)) + j*(1 - (SI.4.12.e + SI.4.13.e) / AP.3.3.e) + k*(SI.4.15.e / SI.4.14.e)
SI.5	d	CSU	(SI.5.4.d)
	d	CSU	a*(1 / (1 + SI.5.1.d)) + b*(SI.5.3.d / SI.5.2.d)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	Procedure	a*(1 / (1 + SI.5.1.e)) + b*(SI.5.3.e / SI.5.2.e) + c*(SI.5.4.e)
SI.6	d	Procedure	1 - ((2 * SI.6.2.d) / (SI.6.1.d * SI.6.3.d))
	e	Procedure	1 - ((2 * SI.6.2.e) / (SI.6.1.e * SI.6.3.e))
SS.1	a	System	a*(SS.1.1.a) + b*(SS.1.2.a) + c*(SS.1.3.a) + d*(SS.1.4.a)
	b	CSCI	a*(SS.1.1.b) + b*(SS.1.2.b) + c*(SS.1.3.b) + d*(SS.1.4.b)
	c	CSCI	a*(SS.1.1.c) + b*(SS.1.2.c) + c*(SS.1.3.c) + d*(SS.1.4.c)
	g	CSCI	a*(SS.1.1.g) + b*(SS.1.2.g) + c*(SS.1.3.g) + d*(SS.1.4.g)
	h	System	a*(SS.1.1.h) + b*(SS.1.2.h) + c*(SS.1.3.h) + d*(SS.1.4.h)
SS.2	a	System	a*(SS.2.1.a) + b*(SS.2.2.a)
	b	CSCI	a*(SS.2.1.b) + b*(SS.2.2.b)
	c	CSCI	a*(SS.2.1.c) + b*(SS.2.2.c)
	g	CSCI	a*(SS.2.1.g) + b*(SS.2.2.g)
	h	System	a*(SS.2.1.h) + b*(SS.2.2.h)
	i	System	(SS.2.2.i)
ST.1	d	CSU	a*(ST.1.4.d) + b*(ST.1.5.d)
	d	Procedure	a*(1 / (1 + ST.1.1.d)) + b*(ST.1.2.d / ST.1.1.d) + c*(1 - (ST.1.3.d / ST.1.1.d))

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	CSU	a*(1 / ST.1.2.e) + b*(ST.1.4.e) + c*(ST.1.5.e)
	e	Procedure	a*(1 / (1 + ST.1.1.e)) + b*(1 - (ST.1.3.e / AP.2.1.e))
ST.2	d	CSU	(ST.2.5.d)
	d	Procedure	a*(1 / ST.2.1.d) + b*(1 / (1 + ST.2.2.d)) + c*(1 / (1 + ST.2.3.d)) + d*(1 / (1 + ST.2.4.d))
	e	CSU	(ST.2.5.e)
	e	Procedure	a*(1 / ST.2.1.e) + b*(1 / (1 + ST.2.2.e)) + c*(1 / (1 + ST.2.3.e)) + d*(1 / (1 + ST.2.4.e))
ST.3	a	System	a*(ST.3.1.a) + b*(ST.3.2.a)
	b	CSCI	a*(ST.3.1.b) + b*(ST.3.2.b)
	c	CSCI	a*(ST.3.1.c) + b*(ST.3.2.c)
	c	CSC	(ST.3.3.c)
	d	CSCI	(ST.3.2.d)
	d	CSU	a*(ST.3.4.d) + b*(ST.3.5.d)
	e	CSU	a*(ST.3.4.e) + b*(ST.3.5.e)
ST.4	d	CSCI	(ST.4.4.d / ES.1.5.d)
	d	CSU	(ST.4.6.d)
	d	Procedure	a*(1 / (1 + AP.2.2.d)) + c*(ST.4.5.d)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	CSCI	a*(1 / (1 + ST.4.3.e)) + b*(ST.4.4.e / ES.1.5.e)
	e	CSU	a*(ST.4.5.e) + b*(ST.4.6.e)
ST.5	d	CSU	a*(ST.5.1.d) + b*(ST.5.2.d) + c*(ST.5.3.d) + d*(ST.5.4.d)
	e	CSU	a*(ST.5.1.e) + b*(ST.5.2.e) + c*(ST.5.3.e) + d*(ST.5.4.e)
SY.1	a	System	a*(SY.1.1.a) + b*(SY.1.2.a) + c*(SY.1.3.a) + d*(SY.1.4.a)
	b	CSCI	a*(SY.1.1.b) + b*(SY.1.2.b) + c*(SY.1.3.b) + d*(SY.1.4.b)
	c	CSCI	a*(SY.1.1.c) + b*(SY.1.2.c) + c*(SY.1.3.c) + d*(SY.1.4.c)
	g	CSCI	a*(SY.1.1.g) + b*(SY.1.2.g) + c*(SY.1.3.g) + d*(SY.1.4.g)
	h	System	a*(SY.1.1.h) + b*(SY.1.2.h) + c*(SY.1.3.h) + d*(SY.1.4.h)
SY.2	a	System	a*(SY.2.1.a) + b*(SY.2.2.a) + c*(SY.2.3.a)
	b	CSCI	a*(SY.2.1.b) + b*(SY.2.2.b) + c*(SY.2.3.b)
	c	CSCI	a*(SY.2.1.c) + b*(SY.2.2.c) + c*(SY.2.3.c)
	g	CSCI	a*(SY.2.1.g) + b*(SY.2.2.g) + c*(SY.2.3.g)
	h	System	a*(SY.2.1.h) + b*(SY.2.2.h) + c*(SY.2.3.h)
SY.3	a	System	a*(SY.3.1.a) + b*(SY.3.2.a) + c*(SY.3.3.a)
	b	CSCI	a*(SY.3.1.b) + b*(SY.3.2.b) + c*(SY.3.3.b)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	c	CSCI	(SY.3.3.c)
	f	CSCI	(SY.3.2.f)
	g	CSCI	a*(SY.3.1.g) + b*(SY.3.2.g) + c*(SY.3.3.g)
	h	System	a*(SY.3.1.h) + b*(SY.3.2.h) + c*(SY.3.3.h)
SY.4	a	System	a*(SY.4.1.a) + b*(SY.4.2.a) + c*(SY.4.3.a)
	b	CSCI	a*(SY.4.1.b) + b*(SY.4.2.b) + c*(SY.4.3.b)
	c	CSCI	a*(SY.4.1.c) + b*(SY.4.2.c) + c*(SY.4.3.c)
	g	CSCI	a*(SY.4.1.g) + b*(SY.4.2.g) + c*(SY.4.3.g)
	h	System	a*(SY.4.1.h) + b*(SY.4.2.h) + c*(SY.4.3.h)
SY.5	a	System	(SY.5.1.a)
	b	CSCI	(SY.5.1.b)
	c	CSCI	(SY.5.1.c)
TC.1	b	CSCI	(TC.1.1.b)
	c	CSCI	(TC.1.1.c)
	c	CSC	(TC.1.2.c)
	d	CSC	(TC.1.1.d)

Appendix A. Metric Scoring Formulas Based on Metric Elements
by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	d	CSU	(TC.1.2.d)
TN.1	a	System	a*(TN.1.1.a) + b*(TN.1.2.a) + c*(TN.1.3.a) + d*(TN.1.4.a)
	b	CSCI	a*(TN.1.1.b) + b*(TN.1.2.b) + c*(TN.1.3.b) + d*(TN.1.4.b)
	d	CSCI	a*(TN.1.1.d) + b*(TN.1.2.d) + c*(TN.1.3.d) + d*(TN.1.4.d)
	g	CSCI	a*(TN.1.1.g) + b*(TN.1.2.g) + c*(TN.1.3.g) + d*(TN.1.4.g)
	h	System	a*(TN.1.1.h) + b*(TN.1.2.h) + c*(TN.1.3.h) + d*(TN.1.4.h)
VR.1	a	System	(OP.1.15.a)
	b	CSCI	(OP.1.15.b)
	c	CSCI	(OP.1.15.c)
	g	CSCI	(OP.1.15.g)
	h	System	(OP.1.15.h)
VS.1	a	System	(VS.1.4.a)
	b	CSCI	(VS.1.4.b)
	c	CSC	(VS.1.4.c)
	d	Procedure	a*(VS.1.2.d / VS.1.1.d) + b*(VS.1.4.d / VS.1.3.d)
	e	Procedure	a*(VS.1.2.e / VS.1.1.e) + b*(VS.1.4.e / VS.1.3.e)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	f	CSC	(VS.1.4.f / VS.1.3.f)
	g	CSCI	(VS.1.4.g) / (VS.1.3.g)
	h	System	(VS.1.4.h) / (VS.1.3.h)
VS.2	a	System	(VS.2.2.a)
	b	CSCI	(VS.2.2.b)
	c	CSCI	(VS.2.2.c)
	d	CSCI	(VS.2.2.d / VS.2.1.d)
	f	CSC	(VS.2.2.f) / (VS.2.1.f)
	g	CSCI	(VS.2.2.g)
	h	System	(VS.2.2.h)
VS.3	a	System	a*(VS.3.1.a) + b*(VS.3.2.a) + c*(VS.3.3.a)
	b	CSCI	a*(VS.3.1.b) + b*(VS.3.2.b) + c*(VS.3.3.b)
	c	CSCI	a*(VS.3.1.c) + b*(VS.3.2.c) + c*(VS.3.3.c)
	d	CSCI	(VS.3.1.d)
	d	CSC	(VS.3.3.d)
	d	CSU	(VS.3.2.d)

Appendix A. Metric Scoring Formulas Based on Metric Elements by Architecture Level and Phase

METRIC	DCF	LEVEL	METRIC ELEMENTS FORMULA
	e	CSC	(VS.3.2.e)
	e	CSU	(VS.3.3.e)
	f	CSC	(VS.3.3.f)
	f	CSU	(VS.3.2.f)
	g	CSCI	a*(VS.3.1.g) + b*(VS.3.2.g) + c*(VS.3.3.g)
	h	System	a*(VS.3.1.h) + b*(VS.3.2.h) + c*(VS.3.3.h)
VS.4	a	System	(VS.4.1.a)
	b	CSCI	(VS.4.1.b)
	c	CSCI	(VS.4.1.c)
	d	CSC	(VS.4.1.d)
	f	CSC	(VS.4.1.f)
	g	CSCI	(VS.4.1.g)
	h	System	(VS.4.1.h)

APPENDIX B -- CRITERION SCORING FORMULAS

Introduction

This appendix presents the formulas used to calculate criterion scores based on metric calculations. These formulas can be used across all levels of application (DCFs A through I) and across the various levels of application to the system architecture (computer software units, computer software components, computer software configuration items, and the system itself).

To use the formulas, select the criterion desired. Calculate metric scores for the particular level and phase using either the equations of Appendix A. Use these results in the formulas for this appendix. If a particular metric is not available or applicable to the phase or level selected, omit it and its coefficient from the equation.

The following applies to each of the formulas listed:

* indicates a multiplication is to be performed

+ indicates an addition is to be performed

(AA.n) represents the value for a particular metric

a,b,c,d... are multiplying coefficients which should sum to a value of 1 in each equation. Weighting can change the relative values of these coefficients, but they should always sum to 1. If a metric has been omitted as N/A, then its coefficient becomes 0. The default values of the coefficients are set to produce an average, with each coefficient equal to the others.

APPENDIX B -- CRITERION SCORING FORMULAS

CRITERION		FORMULA
ACCURACY	AC	(AC.1)
ANOMALY MANAGEMENT	AM	a*(AM.1) + b*(AM.2) + c*(AM.3) + d*(AM.4) + e*(AM.5) + f*(AM.6) + g*(AM.7)
APPLICATION INDEPENDENCE	AP	a*(AP.1) + b*(AP.2) + c*(AP.3) + d*(AP.4) + e*(AP.5)
AUGMENTABILITY	AT	a*(AT.1) + b*(AT.2) + c*(AT.3) + d*(AT.4)
AUTONOMY	AU	a*(AU.1) + b*(AU.2)
COMMONALITY	CL	a*(CL.1) + b*(CL.2) + c*(CL.3)
COMPLETENESS	CP	(CP.1)
CONSISTENCY	CS	a*(CS.1) + b*(CS.2)
DISTRIBUTEDNESS	DI	(DI.1)
DOCUMENT ACCESSIBILITY	DO	a*(DO.1) + b*(DO.2)
EFFECTIVENESS -- COMMUNICATION	EC	(EC.1)

APPENDIX B -- CRITERION SCORING FORMULAS

CRITERION		FORMULA
EFFECTIVENESS -- PROCESSING	EP	a*(EP.1) + b*(EP.2)
EFFECTIVENESS -- STORAGE	ES	(ES.1)
FUNCTIONAL OVERLAP	FO	(FO.1)
FUNCTIONAL SCOPE	FS	a*(FS.1) + b*(FS.2) + c*(FS.3)
GENERALITY	GE	a*(GE.1) + b*(GE.2)
INDEPENDENCE	ID	a*(ID.1) + b*(ID.2)
MODULARITY	MO	a*(MO.1) + b*(MO.2)
OPERABILITY	OP	a*(OP.1) + b*(OP.2) + c*(OP.3)
RELIABILITY	RE	(RE.1)
SELF-DESCRIPTIVENESS	SD	a*(SD.1) + b*(SD.2) + c*(SD.3)
SIMPLICITY	SI	a*(SI.1) + b*(SI.2) + c*(SI.3) + d*(SI.4) + e*(SI.5) + f*(SI.6)
SYSTEM ACCESIBILITY	SS	a*(SS.1) + b*(SS.2)

APPENDIX B -- CRITERION SCORING FORMULAS

CRITERION		FORMULA
SYSTEM CLARITY	ST	a*(ST.1) + b*(ST.2) + c*(ST.3) + d*(ST.4) + e*(ST.5)
SYSTEM COMPATIBILITY	SY	a*(SY.1) + b*(SY.2) + c*(SY.3) + d*(SY.4) + e*(SY.5)
TRACEABILITY	TC	(TC.1)
TRAINING	TN	(TN.1)
VIRTUALITY	VR	(VR.1)
VISIBILITY	VS	a*(VS.1) + b*(VS.2) + c*(VS.3) + d*(VS.4)

APPENDIX C -- FACTOR SCORING FORMULAS

Introduction

This appendix presents the formulas used to calculate factor scores based on criteria calculations. These formulas can be used across all levels of application (DCFs A through I) and across the various levels of application to the system architecture (computer software units, computer software components, computer software configuration items, and the system itself).

To use the formulas, select the factor desired. Calculate criteria scores for the particular level and phase using the equations of Appendix B. Use these results in the formulas for this appendix. If a particular criterion is not available or applicable to the phase or level selected, omit it and its coefficient from the equation.

The following applies to each of the formulas listed:

*	indicates a multiplication is to be performed
+	indicates an addition is to be performed
(AA)	represents the value for a particular criterion
a,b,c,d...	are multiplying coefficients which should sum to a value of 1 in each equation. Weighting can change the relative values of these coefficients, but they should always sum to 1. If a criterion has been omitted as N/A, then its coefficient becomes 0. The default values of the coefficients are set to produce an average, with each coefficient equal to the others.

APPENDIX C -- FACTOR SCORING FORMULAS

FACTOR	FORMULA
EFFICIENCY	a*(EC) + b*(EP) + c*(ES)
INTEGRITY	(SS)
RELIABILITY	a*(AC) + b*(AM) + c*(SI)
SURVIVABILITY	a*(AM) + b*(AU) + c*(DI) + d*(MO) + e*(RE)
USABILITY	a*(OP) + b*(TN)
CORRECTNESS	a*(CP) + b*(CS) + c*(TC)
MAINTAINABILITY	a*(CS) + b*(MO) + c*(SD) + d*(SI) + e*(VS)
VERIFIABILITY	a*(MO) + b*(SD) + c*(SI) + d*(VS)
EXPANDABILITY	a*(AT) + b*(GE) + c*(MO) + d*(SD) + e*(SI) + f*(VR)
FLEXIBILITY	a*(GE) + b*(MO) + c*(SD) + d*(SI)
INTEROPERABILITY	a*(CL) + b*(FO) + c*(ID) + d*(MO) + e*(SY)
PORTABILITY	a*(ID) + b*(MO) + c*(SD)
REUSABILITY	a*(AP) + b*(DO) + c*(FS) + d*(GE) + e*(ID) + f*(MO) + g*(SD) + h*(SI) + i*(ST)

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.